Tree-based statistical learning for modeling genetic risk scores and identifying gene–gene and gene–environment interactions

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Michael Lau
aus Neuss

Düsseldorf, Februar 2024

aus dem Mathematischen Institut
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Dr. Holger Schwender
   Heinrich-Heine-Universität Düsseldorf

2. PD Dr. Tamara Schikowski
   IUF – Leibniz-Institut für umweltmedizinische Forschung GmbH

3. Prof. Dr. Klaus Jung
   Stiftung Tierärztliche Hochschule Hannover

Tag der mündlichen Prüfung: 9. Juli 2024

Parts of this dissertation have been published:

1. M. Lau, C. Wigmann, S. Kress, T. Schikowski, and H. Schwender. Evaluation of tree-based statistical learning methods for constructing genetic risk scores. *BMC Bioinformatics*, 23:97, 2022. doi:10.1186/s12859-022-04634-w.

2. M. Lau, S. Kress, T. Schikowski, and H. Schwender. Efficient gene–environment interaction testing through bootstrap aggregating. *Scientific Reports*, 13:937, 2023. doi:10.1038/s41598-023-28172-4.

3. M. Lau, T. Schikowski, and H. Schwender. logicDT: a procedure for identifying response-associated interactions between binary predictors. *Machine Learning*, 113(2):933–992, 2024. doi:10.1007/s10994-023-06488-6.

Abstract

Genetic risk scores (GRS) summarize (parts of) the genetic makeup of individuals with regard to a specific phenotype such as a disease status. GRS can be used for personal risk assessment or for deriving biological mechanisms involved in the development of the considered phenotype. It is well known that genetic variants do not have to independently influence the considered outcome but they might also interact with each other. GRS are commonly constructed using linear approaches such as the elastic net or aggregating individual effect estimates of genetic variants. Linear approaches, however, do not incorporate such gene–gene interaction effects, unless prior knowledge about which predictors might interact is available, which is typically not the case in genetic epidemiology.

Therefore, tree-based statistical learning methods that are able to autonomously detect and incorporate interaction effects are investigated for their ability to construct GRS in this thesis. More precisely, variants of random forests and logic regression are evaluated against the elastic net. Simulation studies as well as a real data application show that these tree-based methods are able to outperform the elastic net in terms of the induced predictive ability.

Genetic risk factors can also interact with environmental risk factors in the development of complex phenotypes. Standard statistical tests for testing the presence of such a gene–environment (GxE) interaction effect either do not properly model the genetic risk factors or suffer from reduced statistical power due to splitting the available data into training data sets for constructing a GRS and test data sets for statistically testing the GxE interaction effect to avoid overfitting. Therefore, a novel GxE interaction test is designed that utilizes bagging (bootstrap aggregating) and OOB (out-of-bag) predictions to both construct a GRS model and subsequently test the GxE interaction using the complete data set. Moreover, it is proposed to employ random forests as the GRS construction procedure, as random forests yielded high predictive performances in the first part of this dissertation due to flexibly modeling arbitrary effects. Empirical evaluations show that the proposed GxE interaction test yields a high statistical power while controlling the type I error rate.

A notable shortcoming of the ensemble tree methods random forests and logic regression with bagging, that yield GRS with comparably strong associations with the outcome, is their lack of interpretability, i.e., contrary to elastic net models, it can no longer be easily understood how predictions are composed and which predictors influence the outcome in which interplay and magnitude. Hence, a novel statistical learning method is developed that constructs a single decision tree that can split on single predictors or Boolean conjunctions/interactions of multiple predictors. This procedure, therefore, captures gene–gene interactions on split level, and moreover, incorporates GxE interactions by fitting regression models in the decision tree leaves. This statistical learning method is accompanied by a framework for measuring the importance of predictors and interactions between predictors. In simulation studies and real data applications, it is shown that this new method yields strongly predictive and interpretable models.

Acknowledgments

First of all, I would like to thank my supervisor, Holger Schwender. He always gave me the freedom and support I needed for my work. Through him, I received the opportunity to join the RTG and enjoy a wonderful research atmosphere. Thank you, Holger, you made this possible and made me evolve both scientifically and personally. I would also like to thank my co-supervisor, Tamara Schikowski. Sara, Claudia (who work in Tamara's working group at the IUF), and Tamara helped me make my theoretical research become more practical by providing real data to be analyzed.

I am glad to have been part of the RTG, where I got to know many amazing people (especially Timur and Akin) who always encourage each other.

I want to thank my wife, Beatrix, for her endless support in every aspect. I am also forever grateful to my father, Jürgen, who encouraged me to pursue my dreams for my whole life. I deeply appreciate the support of my aunt and uncle, Gaby and Kurt, who always believed in me.

Last but not least, I would like to thank my dearest friends, Kevin, Mai-Han, Pascal, René, and Tim, who I shared countless unforgettable memories with.

This work has been supported by the Research Training Group "Biostatistical Methods for High-Dimensional Data in Toxicology" (RTG 2624, Project R3) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation – Project Number 427806116). Computational infrastructure and support were provided by the Centre for Information and Media Technology at the Heinrich Heine University Düsseldorf.

Contents

Abbreviations

BMI	Body mass index
CART	Classification and regression tree
DNA	Deoxyribonucleic acid
ESCAPE	European study of cohorts for air pollution effects
EWAS	Environment-wide association study
GAM	Generalized additive model
GLM	Generalized linear model
GLMM	Generalized linear mixed-effects model
GRS	Genetic risk score
GWAS	Genome-wide association study
GxE	Gene–environment
HLA	Human leukocyte antigen
iid	Independent and identically distributed
LD	Linkage disequilibrium
logicDT	Logic decision tree
mRNA	Messenger ribonucleic acid
OOB	Out-of-bag
PRS	Polygenic risk score
RNA	Ribonucleic acid
SALIA	Study on the influence of air pollution on lung function, inflammation, and aging
SBERIA	Set-based gene–environment interaction test
SHAP	Shapley additive explanation
SNP	Single nucleotide polymorphism
VIM	Variable importance measure
XGBoost	Extreme gradient boosting

Chapter 1

Introduction

The manifestation of complex phenotypes such as disease statuses or quantitative biomarkers may be influenced by many types of risk factors, such as genetic makeup, exposure to environmental pollutants, or lifestyle [Wild, 2005]. In genetic epidemiology, point mutations in the DNA, i.e., single substitutions of base pairs, are measured and analyzed as a common type of genetic mutation [George Priya Doss et al., 2008]. However, the human genome consists of approximately 3.2 billion base pairs [International Human Genome Sequencing Consortium, 2001] so that usually not all positions in the DNA are analyzed but only variants that occur with a frequency of at least one percent. The human genome contains approximately 85 million of these more common single base-pair mutations which are called *SNPs* (*single nucleotide polymorphisms*) [The 1000 Genomes Project Consortium, 2015]. Therefore, analyzing the effect of SNPs on the development of a considered phenotype creates a high-dimensional modeling problem. Moreover, epidemiological studies such as cohort studies or case-control studies often do not consist of more than a few thousand observations. Hence, the considered task of modeling genetic risk factors typically involves a data set where the sample size is a small fraction of the number of predictors ($n \ll p$).

Due to this problem complexity, SNPs are often analyzed individually, i.e., statistically testing the (marginal) effect of each considered SNP on the considered phenotype and assessing its effect size, as in GWAS (genome-wide association studies) [Uffelmann et al., 2021]. This approach is computationally efficient and allows sharing summary statistics about the individual SNPs regarding the considered outcome that can be used in independent studies.

These individually estimated effect sizes may be used for constructing *genetic risk scores* (*GRS*) (also called *polygenic risk scores* (*PRS*)) that summarize multiple genetic variants such as SNPs of individuals in a single statistic considering a specific outcome [Lewis and Vassos, 2017]. Constructing accurate GRS allows unveiling underlying biological mechanisms involved in the development of the considered phenotype. Moreover, GRS can be also applied in a clinical context for predicting disease risks and potentially advising preventive measures to reduce the personal risk in precision medicine [Torkamani et al., 2018, Lewis and Vassos, 2020, Wray et al., 2021]. Traditionally, GRS are constructed as weighted sums of SNPs that are used to estimate the genetic liability to a trait [Lewis and Vassos, 2017, Choi et al., 2020]. However, these linear models with individually estimated effect coefficients do not take interactions or correlations between SNPs into account which limits the modeling capability.

Therefore, one central idea of this dissertation is to generalize the GRS definition from linear models to arbitrary functions that assign a considered set of SNPs a risk estimate. Tree-based statistical learning methods construct functions that can autonomously detect and include interaction effects and are investigated and refined for constructing GRS in this work. However, these flexible statistical learning procedures can be computationally intensive so that it might not be feasible to incorporate all SNPs at once. For example, the software implementation of the tree-based statistical learning procedure logic regression [Ruczinski et al., 2003, Kooperberg and Ruczinski, 2023] allows a maximum of only 1,000 predictors.

To reduce the dimensionality of the problem without losing substantial information, SNPs are often pruned based on LD (linkage disequilibrium), i.e., based on correlation structures between SNPs, by identifying correlation clusters and reducing each cluster to a representative SNP [see, e.g., Purcell et al., 2007]. Alternatively or in addition, subsets of SNPs, e.g., all SNPs contained in a specific gene, chromosome, or genetic pathway or all SNPs that showed significant associations with the considered outcome in prior analyses, can be considered to conduct more specific analyses with lower dimensionality. Reducing the number of SNPs to not more than a few hundred allows employing more sophisticated modeling procedures that are able to capture arbitrarily complex interaction effects for constructing GRS.

The biological background about SNPs, environmental risk factors, and epidemiological studies for assessing their influences on the development of complex diseases is provided in the following sections. Afterward, joint modeling of SNPs in GRS is discussed in detail.

1.1 Genetics

Almost every cell in the human body contains a complete copy of all genetic information of an individual, the *genome*. The human genome is structured as *chromosome* pairs, where, for each pair, one chromosome is inherited from the mother and one chromosome is inherited from the father. Each chromosome contains two intertwined *DNA* (*deoxyribonucleic acid*) strands that are chains of *nucleotides*. Nucleotides consist of a phosphate group, a deoxyribose sugar, and a *nitrogen base*, which can be either adenine, thymine, cytosine, or guanine. It is sufficient to know the nitrogen bases in one DNA strand, since the other DNA strand contains the complementary nitrogen bases so that adenine is always connected to thymine and cytosine is always connected to guanine [Graw, 2015]. Genetic *loci* (singular *locus*) describe specific positions in the genome.

The complete genetic material of an individual is known as its *genotype*. For analyzing the influence of genetic components on the manifestation of *phenotypes*, i.e., any observable characteristic of an individual such as a disease status or a biomarker, the genotypes of multiple individuals are measured in studies. To obtain statistically useful predictors, specific types of genetic *mutation*/*variation* are considered. Types of genetic mutation include *point mutations*, where single base-pairs are substituted, inserted, or deleted and *chromosomal mutations*, where parts of a chromosome are deleted, inverted, repeated, or relocated to another chromosome [Clancy, 2008].

In this work, single base-pair substitutions are considered. More precisely, the influence of *SNPs* (*single nucleotide polymorphisms*), where the less common base-pair is occurring in at least one percent of the reference population, is investigated. SNPs are defined by a *major allele*, i.e., a base-pair that is more common in the reference population, and a *minor allele*, i.e., a base-pair that is less common in the reference population. As the human is a *diploid* organism, i.e., as the human carries two copies of each chromosome, the minor (or the major) allele may be present on no, exactly one, or both chromosome(s). Thus, SNPs are coded as $\{0, 1, 2\}$, counting the number of minor allele occurrences with respect to both chromosome copies.

For measuring SNPs of an individual, first, a saliva or blood sample containing DNA is collected. Next, the contained DNA is isolated and amplified (i.e., duplicated) to obtain more DNA molecules that can be analyzed. DNA *microarrays* can then be used to assess the SNPs in the prepared sample [Graw, 2015]. DNA microarrays contain short reference DNA sequences that include the considered loci with known alleles and that bind to the sample DNA if the reference DNA and the sample DNA are complementary. The sample DNA is, furthermore, labeled with fluorescent dyes to emit light signals if the sample DNA binds to the reference DNA in the microarray. These light signals are captured to deduce the SNPs that can be measured with the employed microarray, i.e., the SNPs for which corresponding reference DNA is contained in the microarray [Graw, 2015].

SNPs can exhibit different *modes of inheritance* regarding a considered phenotype [Scherer et al., 2021]. A SNP is exhibiting a *dominant* mode of inheritance, if the effect on the phenotype is present if the minor allele is occurring on at least one chromosome copy, i.e., if $\text{SNP} > 0$. Furthermore, a SNP is exhibiting a *recessive* mode of inheritance, if the effect on the phenotype is present if the minor allele is occurring on both chromosomes copies at once, i.e., if $\text{SNP} = 2$. Different modes of inheritance are also possible. However, using the two binary variables $\mathbb{1}(\text{SNP} > 0)$ and $\mathbb{1}(\text{SNP} = 2)$, any other effect type may be modeled due to $\text{SNP} = \mathbb{1}(\text{SNP} > 0) + \mathbb{1}(\text{SNP} = 2)$.

LD (*linkage disequilibrium*) describes the deviation of observed joint allele frequencies from expected joint allele frequencies under the assumption that the considered genetic loci are statistically independent [Slatkin, 2008]. Hence, LD can be interpreted as a measure of correlation between different genetic loci. Genetic loci that are physically close to each other tend to be in higher LD than loci that are far away from each other [Ardlie et al., 2002]. LD structures can be utilized to impute unmeasured SNPs by inferring the most plausible *haplotype*, i.e., combination of genetic variants, given the measured SNPs and a reference panel of observed haplotypes [Khankhanian et al., 2015, Shi et al., 2019]. Conversely, since two SNPs can be in very high LD (e.g., exhibiting an empirical correlation $r > 0.9$), SNPs are commonly pruned based on their LD, i.e., correlation clusters of SNPs are reduced to one representative SNP per cluster to reduce the number of genetic variables without losing too much information [Purcell et al., 2007, Calus and Vandenplas, 2018, Hüls and Czamara, 2020].

Certain parts of the DNA strands are *genes* (approximately one percent), which means that, according to the central dogma of molecular biology, these DNA regions encode *proteins*, by genes being transcribed into *mRNA* (*messenger ribonucleic acid*) and mRNA being translated into proteins [Graw, 2015]. Proteins consist of amino acids and are crucial for the structure and function of cells. Since a gene modification can lead to a protein modification, especially SNPs in gene regions are commonly analyzed.

1.2 Environmental risk factors

Not only genetic risk factors but also exposure to environmental risk factors is an important component in the manifestation of phenotypes. As an environmental counterpart to the genome, the *exposome* captures all human encounters with environmental risk factors [Wild, 2005].

An important type of environmental exposure is the exposure to air pollutants that are, e.g., caused by traffic. For example, it has already been shown that air pollution exposure decreases lung function [Schikowski et al., 2005] and increases the disease risk of type 2 diabetes mellitus [Eze et al., 2015]. Therefore, in this work, exposures to the air pollutants NO_2 (nitrogen dioxide), NO_x (nitrogen monoxide NO and nitrogen dioxide NO_2), $PM_{2.5}$ (particulate matter with an aerodynamic diameter of less than 2.5 µm), PM_{10} (particulate matter with an aerodynamic diameter of less than 10 µm), PM_{coarse} (particulate matter with diameters between 2.5 µm and 10 µm), and $PM_{2.5\text{ absorbance}}$ (reflectance of $PM_{2.5}$ filters [see, e.g., Eeftens et al., 2015]) are studied as potentially explanatory variables in addition to genetic factors.

Exposure to environmental risk factors might be influenced by other variables such as lifestyle indicators. These other variables might *confound*, i.e., distort, the influence of the environmental variable on the outcome [Pourhoseingholi et al., 2012, Johnston et al., 2018]. For example, the socioeconomic status of an individual might influence the residential area, and therefore, also the air pollution exposure (e.g., due to less traffic in the residential area). Moreover, the socioeconomic status might influence the risk of a specific disease (e.g., due to being able to afford a healthier diet). Therefore, in this example, estimated effects of air pollution exposure on the disease risk might not reflect the true causal effects if the socioeconomic status is not explicitly taken into account due to the estimated air pollution effects partially being indirect socioeconomic effects in this case [Hajat et al., 2021]. Hence, if causal effects should be investigated and it is suspected that the relationship between the considered predictors and the outcome might be confounded, potential confounding variables should be included in the model fitting procedure to adjust for this phenomenon [Pourhoseingholi et al., 2012].

1.3 Epidemiological studies

For statistically assessing the influence of SNPs and environmental risk factors on the development of diseases, epidemiological studies such as genetic association studies are conducted in which the considered SNPs, the considered phenotype, as well as other individual characteristics of study participants (such as environmental exposures or lifestyle indicators) are measured [Hirschhorn et al., 2002]. An important study type are *GWAS* (*genome-wide association studies*), where SNPs from the whole genome are analyzed [Uffelmann et al., 2021]. *EWAS* (*environment-wide association studies*) are the environmental counterpart to GWAS that analyze the effects of the exposome on considered phenotypes [Patel et al., 2010].

Common study designs include *case-control studies*, where study participants are chosen based on their disease statuses, and *cohort studies*, where study participants are followed over time so that, e.g., initially no study participant might have a considered disease but disease incidences could be observed in follow-up examinations [Woodward, 2013].

In this thesis, existing and newly proposed approaches for constructing GRS are also evaluated using data from the German *SALIA* (*study on the influence of air pollution on lung function, inflammation, and aging*) cohort study [Schikowski et al., 2005]. The participants of the SALIA study were recruited in the period of 1985–1994 from highly and less industrialized areas in North-Rhine Westphalia, Germany. At its baseline examination, the study included 4874 women that were between 54 and 55 years old. In a follow-up clinical examination that was conducted in the period of 2007–2010, SNPs of study participants were measured using the Axiom Precision Medicine Research Array GRCh37/hg19 (Affymetrix, Santa Clara, CA, USA). Moreover, individual exposures to air pollutants such as NO_2 were estimated for different time points using land-use regression models, that utilize geographical data and measurements from fixed monitoring sites, as part of the ESCAPE (European study of cohorts for air pollution effects) project [Eeftens et al., 2012, Beelen et al., 2014].

Among many disease outcomes, the presence of rheumatic diseases has been collected and is used as the outcome of interest in this work. Thus, a data set consisting of over 500 observations is analyzed that contains both SNP data and the presence of rheumatic diseases.

The development of rheumatic diseases involves a complex interplay of genetic and non-genetic risk factors [Kirino and Remmers, 2015]. Due to rheumatoid arthritis being the most common rheumatic disease besides osteoarthritis [Sangha, 2000, Vanhoof et al., 2002, Jokar and Jokar, 2018], SNP selections are performed considering prior studies involving rheumatoid arthritis. For example, SNPs from the HLA (human leukocyte antigen) class II complex are analyzed, as genetic loci from this complex showed significant associations with rheumatoid arthritis in prior studies [Zanelli et al., 2000, Kampstra and Toes, 2017].

1.4 Constructing genetic risk scores

GRS are usually constructed as weighted sums of SNPs using *external weights* that are obtained through summary statistics from independent association studies. Therefore, predictions on the complete considered data set can be made using external weights, since the model, i.e., the SNP coefficients/weights, was already determined beforehand in an independent study. External weights might, however, not be available for the considered phenotype, genomic region, or population type [Hüls et al., 2017b]. Hence, in this case, *internal weights* have to be estimated using the available data set. For that reason, to avoid overfitting, the considered data set has to be split into a training data set for estimating the weights and a test data set for performing predictions and assessing the predictive performance of the GRS. Internal weights can also be obtained by employing multiple regression procedures such as linear or logistic regression or regularized variants such as the elastic net [Zou and Hastie, 2005, Hüls et al., 2017a, Privé et al., 2019].

However, interaction effects between SNPs are not captured by conventional GRS construction approaches such as GLMs (generalized linear models) unless prior knowledge about which genetic loci might interact with each other is available which is usually not the case, as the number of all possible interaction terms increases exponentially with the number of predictors (see Section 1.6).

Instead of only considering linear models, GRS can be also, more generally, seen as functions $\varphi : \boldsymbol{\mathcal{X}} \to \mathbb{R}$ that assign a set of considered SNPs in the p-dimensional space $\boldsymbol{\mathcal{X}} = \{0, 1, 2\}^p$ a quantitative risk estimate for the considered phenotype. Therefore, if the considered SNPs are random variables $\begin{pmatrix} X_1 & \dots & X_p \end{pmatrix}^T =: \boldsymbol{X} \in \boldsymbol{\mathcal{X}}$ and the considered phenotype is a random variable $Y \in \mathcal{Y} \subseteq \mathbb{R}$, the problem of constructing a GRS can be interpreted as a *supervised statistical learning problem* in which the true regression function $\mu(\boldsymbol{x}) := \mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]$ has to be estimated [Hastie et al., 2009]. For this purpose, it is assumed that a training data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, i.e., a random sample from $\mathbb{P}_{(\boldsymbol{X},Y)}^{\otimes n}$ which is a data set of n

observations from independent and identically distributed (iid) random vectors $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n) \overset{\text{iid}}{\sim} \mathbb{P}_{(\boldsymbol{X},Y)}$, is available.

1.5 Linear statistical learning methods

GLMs are also statistical learning methods that yield models

$$g(\mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p, \tag{1}$$

where g is a link function mapping the outcome domain to $\mathbb{R}$, β_0 is an intercept, and $\beta_1, \ldots, \beta_p$ are regression coefficients. These linear models are easily interpretable, since effect sizes can be directly read off using the regression coefficients $\beta_1, \ldots, \beta_p$. Moreover, linear hypotheses such as $H_0 : \beta_j = 0$ vs. $H_1 : \beta_j \neq 0$ can be statistically tested, e.g., using Wald tests. However, for $p \geq n$, the maximum-likelihood estimates for $\beta_0, \ldots, \beta_p$ are not unique [Hastie et al., 2015].

Thus, an important extension of GLMs are *regularization* techniques that introduce additional constraints to the optimization problem. Well-established regularization methods include the *lasso* [Tibshirani, 1996] and *ridge regression* [Hoerl and Kennard, 1970]. Instead of directly maximizing the model likelihood function, these procedures consider the constrained optimization problem

$$\min_{(\beta_0, \boldsymbol{\beta})} -\frac{1}{n}\ell(\beta_0, \boldsymbol{\beta}) \quad \text{such that} \quad ||\boldsymbol{\beta}||_q^q = \sum_{j=1}^{p} |\beta_j|^q \leq R, \tag{2}$$

where $\boldsymbol{\beta} := \begin{pmatrix}\beta_1 & \ldots & \beta_p\end{pmatrix}^T$ is the vector of regression coefficients, ℓ is the log-likelihood function, $R \geq 0$ is a parameter controlling the effect sizes, and $q \geq 0$ defines the type of norm used for regularization.

Usually, the optimization problem from Equation (2) is phrased in its Lagrangian form

$$\min_{(\beta_0, \boldsymbol{\beta})} \left\{ -\frac{1}{n}\ell(\beta_0, \boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_q^q \right\}, \tag{3}$$

where $\lambda \geq 0$ is a penalty parameter controlling the regularization strength. For $q \geq 1$, the regularized regression optimization problems from Equations (2) and (3) are equivalent in the sense that for each problem and $R \geq 0$ there is a $\lambda \in [0, \infty]$

leading to the same solution and vice versa [Fu, 1998].

The lasso is a special case of Equations (2) and (3) with $q = 1$ and leads to sparse solutions, i.e., it shrinks regression coefficients of unimportant predictors to zero so that, implicitly, also a variable selection is performed [Tibshirani, 1996]. Ridge regression is also a special case of Equations (2) and (3) with $q = 2$. In contrast to the lasso, ridge regression does not shrink regression coefficients to exactly zero, i.e., it does not perform a variable selection, but ridge regression can properly handle multicollinearity by assigning similar coefficients to highly correlated predictors [Hastie et al., 2015].

For constructing GRS, a compromise between the lasso and the ridge regularizations, the *elastic net* [Zou and Hastie, 2005], is often employed [Hüls et al., 2017a, Privé et al., 2019]. The elastic net considers the optimization problem

$$\min_{(\beta_0,\boldsymbol{\beta})} \left\{ -\frac{1}{n}\ell(\beta_0,\boldsymbol{\beta}) + \lambda \left[\frac{1}{2}(1-\alpha)||\boldsymbol{\beta}||_2^2 + \alpha||\boldsymbol{\beta}||_1 \right] \right\},$$

where $\alpha \in [0, 1]$ is a parameter controlling the balance between the lasso and the ridge regularization. The elastic net combines the advantages of both regularization types by performing a variable selection through the lasso and tending to include or exclude groups of correlated predictors [Hastie et al., 2015]. Moreover, the elastic net yields for $\alpha < 1$ and $\lambda > 0$ a strictly convex optimization problem which induces a unique solution for the regression parameters.

1.6 Interaction effects

GLMs or regularized variants that produce models as in Equation (1) only consider marginal effects of predictors, as the effects, i.e., the coefficients $\beta_1, \dots, \beta_p$, of predictors are constant and do not vary based on other predictors for a new prediction. However, interaction effects between predictors, which are defined in the following, might be also involved in the composition of the outcome.

Definition 1 (Interaction effects [Sorokina et al., 2008]). Two predictors X_i and X_j are said to *interact* with each other considering the prediction function f (e.g., the (transformed) regression function $f(\boldsymbol{X}) = g(\mathbb{E}[Y \mid \boldsymbol{X}])$ for an appropriate link function g), if the effect of X_i depends on X_j, i.e., if $\frac{\partial}{\partial X_i} f(\boldsymbol{X})$ (or finite differences for discrete predictors) depend(s) on X_j, or vice versa.

This definition of interactions is equivalent to an interaction effect between X_i and X_j being present, if the prediction function f cannot be decomposed into a sum $f(\boldsymbol{X}) = f_{\setminus i}(\boldsymbol{X}_{\setminus i}) + f_{\setminus j}(\boldsymbol{X}_{\setminus j})$ in which the first summand does not depend on X_i and the second summand does not depend on X_j [Friedman and Popescu, 2008].

Moreover, this interaction definition can be generalized to interactions of arbitrary order. Predictors $X_{j_1}, \dots, X_{j_k}$ interact with each other, if f cannot be decomposed into a sum of functions

$$f(\boldsymbol{X}) = f_{\setminus j_1}(\boldsymbol{X}_{\setminus j_1}) + \ldots + f_{\setminus j_k}(\boldsymbol{X}_{\setminus j_k})$$

or equivalently (for sufficiently smooth f)

$$\frac{\partial^k}{\partial X_{j_1} \ldots \partial X_{j_k}} f(\boldsymbol{X}) \not\equiv 0.$$

If linear models are considered, interaction effects can be included by recursively modeling the effect coefficients, i.e., $\beta_j = \frac{\partial}{\partial X_j} g(\mathbb{E}[Y \mid \boldsymbol{X}])$ in Equation (1), as linear models of other predictors, e.g., $\beta_2(x_1) = \tilde{\beta}_0 + \tilde{\beta}_1 x_1$ for two predictors. This introduction of interaction effects to GLMs and regularized procedures can be seen as a special case of varying-coefficient models [Hastie and Tibshirani, 1993] and would lead to models

$$\begin{aligned} g(\mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \sum_{j_1=1}^{p} \sum_{j_2=j_1+1}^{p} \gamma_{\{j_1,j_2\}} x_{j_1} x_{j_2} \\ + \sum_{j_1=1}^{p} \sum_{j_2=j_1+1}^{p} \sum_{j_3=j_2+1}^{p} \gamma_{\{j_1,j_2,j_3\}} x_{j_1} x_{j_2} x_{j_3} \\ + \ldots + \gamma_{\{1,2,\ldots,p\}} x_1 x_2 \ldots x_p, \end{aligned}$$

where the coefficient β_j is the main effect of the predictor X_j and the coefficient $\gamma_{\{j_1,\ldots,j_k\}}$ is the interaction effect between the predictors $X_{j_1}, \ldots, X_{j_k}$.

Usually, only main effects are included in GLMs, since the total number of parameters, if all interaction terms are considered, is given by

$$\sum_{k=0}^{p} \binom{p}{k} = 2^p,$$

which yields already for $p = 50$ more than 10^{15} terms and parameters. However, approaches for fitting linear models that can include pairwise interactions $\gamma_{\{i,j\}}x_i x_j$ have been proposed more recently [see, e.g., Lim and Hastie, 2015, Yu et al., 2019]. In this case of only considering second-order interactions, the number of parameters increases quadratically with p.

1.7 Tree-based statistical learning methods

One important class of statistical learning methods that can incorporate interaction effects of arbitrary order without providing prior knowledge are tree-based methods that fit rooted trees from a graph-theoretic point of view. The following graph-theoretic definitions can be also found in Louppe [2014].

Definition 2 (Graph, path, and tree)**.** A (directed) *graph* is a pair $G = (V, E)$, where V is a set of *nodes* and $E \subseteq V \times V$ is a set of *edges*. A *path* in a graph $G = (V, E)$ is a sequence of nodes $(t_1, \ldots, t_m)$ satisfying $(t_i, t_{i+1}) \in E$ for all $i \in \{1, \ldots, m-1\}$ and $t_i \neq t_j$ for all $i \neq j$. A *tree* is a graph $G = (V, E)$, where any two nodes $t_1, t_2 \in V$ are connected by a unique path.

The next definition introduces important types of nodes that are frequently referred to in the context of trees.

Definition 3 (Types of nodes)**.** t_2 is a *child* of t_1 if $(t_1, t_2) \in E$ holds (i.e, if there is an edge from t_1 to t_2). In this case, t_1 is a *parent* of t_2. A node is *internal* if it has at least one child. Otherwise, it is a *terminal* node which is also known as a *leaf.*

In this work, binary trees are exclusively considered, as they are the most common tree type that is produced by tree-based statistical learning procedures.

Definition 4 (Rooted and binary tree)**.** A *rooted* tree is a tree, where one node r has been designated as the *root* of this tree and all edges lead away from the root, i.e., any path $(t_1, \ldots, t_m)$ either starts at the root $(t_1 = r)$ or does not contain the root $(t_i \neq r$ for all $i \in \{1, \ldots, m\})$. A *binary* tree is a rooted tree in which all internal nodes have exactly two children.

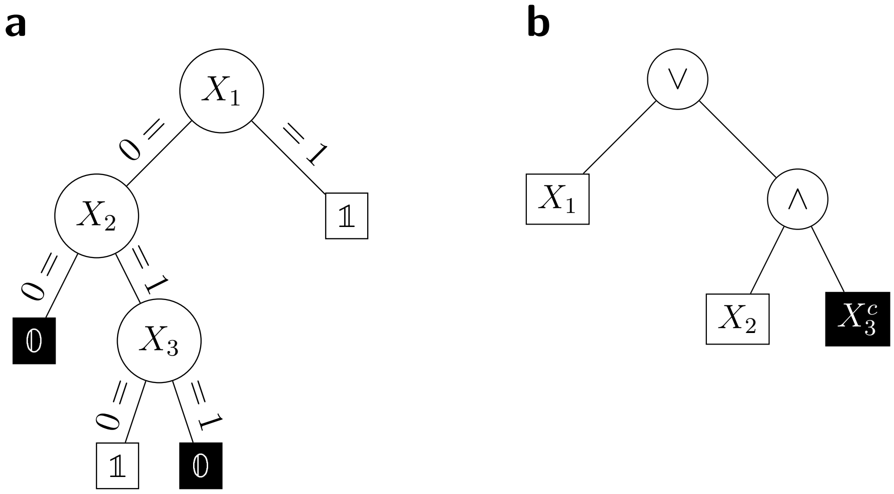

Figure 1: Tree-based models that describe the prediction function $f(\boldsymbol{X}) = X_1 \vee (X_2 \wedge X_3^c)$. In **a**, a decision tree is shown. In **b**, a logic tree is presented. Reproduced from Lau et al. [2024].

1.7.1 Decision trees and random forests

The most widely used tree-based statistical learning method are *decision trees* [Breiman et al., 1984]. Decision trees are binary trees that contain predictors in their internal nodes and prediction values in their leaves. Edges of decision trees are annotated with *splitting rules* based on the predictor in the internal node these edges start from. Splitting rules are specified by triplets (X_j, A, A^c), where $A \subsetneq \mathcal{X}_j$ is a proper subset of the domain of X_j and $A^c = \mathcal{X}_j \setminus A$ is the complement of A with respect to the domain of X_j. Therefore, a splitting rule divides the space of a considered predictor into two disjoint subspaces.

In a decision tree, predictions are computed by evaluating the splitting rules and following the edges based on the considered predictor observation, beginning at the root node and advancing until a leaf is reached that holds the desired prediction value. Figure 1**a** illustrates an exemplary decision tree for binary variables. At the root node, the decision tree asks if the considered observation fulfills $X_1 = 0$ or $X_1 = 1$. If $X_1 = 1$ holds, the prediction value 1 is returned. If $X_1 = 0$ holds, the decision tree asks further for the observed value of X_2. If $X_2 = 0$ holds, the prediction value 0 is obtained. Otherwise, the prediction value also depends on X_3. For $X_3 = 0$, the prediction value 1 is returned, and for $X_3 = 1$, the prediction value 0 is returned.

As decision trees recursively partition the predictor space, and hence, the prediction values depend on sequences of predictors and usually not only single predictors, decision trees naturally incorporate interaction effects as defined in Section 1.6.

Decision trees are usually fitted using greedy algorithms that do not exhaustively investigate all possible trees but consecutively solve local optimization problems such as CART (classification and regression trees) [Breiman et al., 1984] and C4.5 [Quinlan, 1993]. However, due to increasing computational capabilities, decision tree procedures that perform a global optimization have been recently proposed as well [see, e.g., Bertsimas and Dunn, 2017, Aglin et al., 2020, Carrizosa et al., 2021, Demirović et al., 2022].

Decision trees can be used for both classification and regression purposes. In the classification case, leaves can either hold hard classifications, i.e., direct class membership predictions, or soft classifications, i.e., class membership probability estimates [Provost and Domingos, 2003]. In the context of constructing GRS, risk estimates are especially useful.

Single decision trees, however, tend to be unstable which means that small modifications of the training data set induce unproportionally extreme modifications of the fitted decision tree. This issue is mainly caused by the greedy fitting algorithm that locally searches for optimal splitting rules [Murthy and Salzberg, 1995, Li and Belford, 2002].

A popular approach to reducing the variance of decision trees is to create an ensemble of decision trees using *bagging* (*bootstrap aggregating*) [Breiman, 1996]. Bagging fits many different decision trees by providing bootstrap samples (i.e., data sets of size n where each observation has been drawn with replacement from the original data set) as training data sets to the individual decision trees. Predictions are then computed by averaging the predictions of the individual decision trees. This reduces the variance of the ensemble model, since, for M identically distributed random variables $Z_1, \ldots, Z_M$ with positive pairwise correlation ρ—such as predictions $T_1(\boldsymbol{x}), \ldots, T_M(\boldsymbol{x})$ of M individual randomized decision trees for a fixed predictor setting $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$, the variance of the mean is equal to [Hastie et al., 2009]

$$\mathbb{V}\mathrm{ar}\left(\frac{1}{M}\sum_{j=1}^{M} Z_j\right) = \rho \cdot \mathbb{V}\mathrm{ar}(Z_1) + \frac{1-\rho}{M} \cdot \mathbb{V}\mathrm{ar}(Z_1). \tag{4}$$

Thus, for $M \to \infty$, the variance tends to $\rho \cdot \mathbb{V}\mathrm{ar}(Z_1)$. The variance reduction of bagging is, therefore, controlled by the correlation between the individual decision trees and the variance of the single decision trees.

Another useful property of ensemble models fitted using bagging is the *OOB* (*out-of-bag*) prediction mechanism. OOB predictions for a training observation $(\boldsymbol{x}, y) \in \mathcal{D}$ are computed by gathering all submodels of the ensemble that did not use this observation for training and averaging the predictions of these submodels for the considered observation. This means, if the ensemble consists of M submodels $T_1, \ldots, T_M$, the OOB prediction for $(\boldsymbol{x}, y)$ is given by

$$\hat{y}_{\text{OOB}} = \frac{1}{\left|\left\{T_j \mid (\boldsymbol{x}, y) \notin \mathcal{D}_{T_j}, j \in \{1, \ldots, M\}\right\}\right|} \sum_{j=1}^{M} T_j(\boldsymbol{x}) \cdot \mathbb{1}\left((\boldsymbol{x}, y) \notin \mathcal{D}_{T_j}\right),$$

where $\mathcal{D}_{T_j}$ is the training data set/bootstrap sample used for fitting the submodel T_j. OOB predictions allow unbiased predictions of outcomes for observations contained in the training data set that mimic test data set predictions, as the model used for generating these predictions never saw the considered observations before and the common overfitting problem for training data set predictions is avoided. This is especially useful for estimating the error of an ensemble model, for estimating the influence of predictors (see Section 1.10), and for testing the presence of gene–environment interaction effects (see Section 1.9).

Random forests randomize the fitting of decision tree ensembles even more by not only employing bagging but also randomizing the greedy search for splitting rules [Breiman, 2001]. More precisely, random forests do not consider all predictors for creating a splitting rule but draw a random subset of predictors which are further investigated for their splitting performances. The idea is to further decorrelate the individual decision trees to gain a higher variance reduction in Equation (4).

However, it has to be kept in mind that the total error of prediction models consists of its variance and its bias [Györfi et al., 2002], i.e., for a fixed predictor observation $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$, the mean squared error can be decomposed into

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[(\mu(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x}))^2\right] &= \left(\mathbb{E}_{\mathcal{D}}\left[f_{\mathcal{D}}(\boldsymbol{x})\right] - \mu(\boldsymbol{x})\right)^2 + \mathbb{V}\text{ar}_{\mathcal{D}}\left(f_{\mathcal{D}}(\boldsymbol{x})\right) \\ &= \text{Bias}_{\mathcal{D}}(f_{\mathcal{D}}(\boldsymbol{x}))^2 + \mathbb{V}\text{ar}_{\mathcal{D}}\left(f_{\mathcal{D}}(\boldsymbol{x})\right),\end{aligned}$$

where $f_{\mathcal{D}}$ is the prediction model fitted to the training data set $\mathcal{D}$ and the expectations and variances are taken with respect to the training data set $\mathcal{D}$ which is a random sample from $\mathbb{P}_{(\boldsymbol{X},Y)}^{\otimes n}$. Hence, when reducing the variance, attention has to be paid that the bias does not increase too much if the total error shall be minimized.

Another class of decision tree ensembles are boosted decision trees [Friedman, 2001], where a particularly popular implementation is XGBoost (extreme gradient boosting) [Chen and Guestrin, 2016]. Boosted decision trees are fitted iteratively to the gradient of the current prediction error, i.e., addressing so-far unexplained variation in the outcome in each iteration.

1.7.2 Logic regression

Logic regression [Ruczinski et al., 2003] is another established tree-based statistical learning procedure that is exclusively designed for binary predictors and constructs models

$$g(\mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]) = \beta_0 + \beta_1 L_1(\boldsymbol{x}) + \ldots + \beta_M L_M(\boldsymbol{x}),$$

where g is a link function and $L_j(\boldsymbol{x}) \in \{0,1\}$ is a Boolean expression evaluated on the (binary) predictor setting $\boldsymbol{x} \in \{0,1\}^p$. Each Boolean expression that uses the Boolean AND ($\wedge$) and OR ($\vee$) operators, Boolean negations (^{c}), and brackets can be transformed into a *logic tree* and vice versa [Ruczinski et al., 2003]. A logic tree is a binary tree that contains the Boolean AND or the Boolean OR in its inner nodes and predictors or their negations in its leaves. To obtain a Boolean expression from a logic tree and to compute predictions using logic regression models, leaves are combined using their parent nodes that contain Boolean operators and recursively proceeding to combine nodes until the root is reached. The tree structure is useful for depicting logic regression models and is also utilized in the fitting procedure of logic regression.

Figure 1**b** illustrates an exemplary logic tree which is equivalent to the decision tree depicted in Figure 1**a**. The leaves X_2 and X_3^c (X_3 negated) are combined using their parent node $\wedge$ to obtain the Boolean expression $X_2 \wedge X_3^c$. This Boolean expression is further combined with X_1 using the root $\vee$ to obtain the prediction model $X_1 \vee (X_2 \wedge X_3^c)$.

Logic regression models are fitted using a stochastic search algorithm, *simulated annealing* [Kirkpatrick et al., 1983], that is based on Markov chains and identifies asymptotically an optimal solution with probability one, as opposed to a greedy algorithm that might get stuck in a local optimum. However, the success of simulated annealing is based on theoretical convergences of the Markov chains and the temperature parameter that, in practice, cannot be guaranteed, as only finitely many computational resources are available and the search space is usually too big to be fully traversed. For employing simulated annealing in logic regression, *states*, i.e., sets of logic trees, are slightly modified to obtain new states that are evaluated and accepted based on their performances, i.e., based on their training data error.

Similar to single decision trees, single logic regression models can be unstable, i.e., exhibit a high variance, if many predictors influence the outcome or if the signal is weak. Therefore, bagging might be applied to logic regression models to reduce the variance and obtain a more stable ensemble model [Schwender and Ickstadt, 2007]. Logic regression models in a bagging-based ensemble are usually trained using a greedy search, as a greedy search requires less computation time than a simulated-annealing-based search and the variance reduction of bagging alleviates the weaknesses of a greedy search [Murthy and Salzberg, 1995].

1.8 Constructing genetic risk scores using tree-based statistical learning methods

SNPs may not only exhibit main effects on the considered phenotype, but they might also interact with each other in the manifestation of this phenotype [Gilbert-Diamond and Moore, 2011, Ritchie and Van Steen, 2018]. These interactions are known as *gene–gene interactions* and might occur between SNPs in the same gene [Dinu et al., 2012] or between SNPs in different genes [Onay et al., 2006, Xiao et al., 2017].

As discussed earlier in Section 1.6, interaction effects are usually not taken into account when fitting GLMs (including regularized variants). Therefore, the tree-based statistical learning methods previously discussed in Section 1.7 might be a superior alternative to construct GRS, as these methods are able to autonomously capture interaction effects.

For constructing GRS for a binary phenotype such as a disease status, random forests have to fit *probability estimation trees* that hold probability estimates in their leaves (instead of hard classifications) to obtain proper risk estimates [Provost and Domingos, 2003]. For logic regression, all considered predictors have to be binary. Therefore, the SNP predictors coded as $\{0, 1, 2\}$ need to be (biologically meaningfully) divided into two binary predictors for each SNP, in $\mathbb{1}(\text{SNP} > 0)$ so that the effect is present if at least one minor allele is present—coding for a dominant mode of inheritance—and in $\mathbb{1}(\text{SNP} = 2)$ so that the effect is present if the minor allele is present on both chromosomes at once—coding for a recessive mode of inheritance. On the contrary, the original SNP predictors can be directly utilized by the random forests fitting procedure, as decision tree splits for quantitative predictors would split a SNP either on $(\text{SNP}, \{0\}, \{1, 2\})$, which is equivalent to a dominant mode of inheritance, or on $(\text{SNP}, \{0, 1\}, \{2\})$, which is equivalent to a recessive mode of inheritance. The predictions of the fitted tree-based GRS models can then be used as the GRS of an individual in the context of precision medicine or for testing the association of the GRS with the phenotype, i.e., validating and quantifying the predictive performance of the model.

Both random forests and logic regression have already been applied in the analysis of SNP data [Bureau et al., 2005, Kooperberg and Ruczinski, 2005, Chen et al., 2011, Yoo et al., 2012, Dinu et al., 2012, Wright et al., 2016]. For modeling SNPs, Kruppa et al. [2012] and Botta et al. [2014] suggest that random forests might induce superior predictive performances compared to linear approaches. However, in the analyses conducted by Gola et al. [2020] and Badré et al. [2021], where genome-wide GRS are considered instead of GRS for single genes or pathways, random forests did not induce substantially better predictive performances than classic GRS construction approaches. Contrarily, a genome-wide GRS for systemic lupus erythematosus constructed using random forests was able to outperform a linear GRS [Ma et al., 2022].

Therefore, there was a lack of studies investigating under which circumstances—such as effect sizes, sample sizes, intensities of statistical noise, or presence/absence of interaction effects—these tree-based methods should be preferred over classical approaches such as the elastic net in the construction of GRS.

In the master's thesis by Lau [2020], a pilot study was conducted to investigate if the tree-based methods random forests and logic regression are able to induce adequate GRS compared to the elastic net in one simple simulation scenario that considered a linear genetic model in which no interaction effects are present. In this scenario, where the model assumptions of the elastic net are fulfilled, and hence, the interaction detection ability of tree-based methods is not required, random forests and logic regression were able to yield GRS with similar predictive performances compared to the elastic net.

Due to this positive result, the first part of this dissertation (see Lau et al. [2022]/Chapter 2) focuses on extending the simulation scenarios and validating the results in an application to a real data set from the SALIA cohort study (see Section 1.3 for more details about the SALIA study). Moreover, the tree-based methods are further refined to the problem of constructing GRS. Instead of considering classification or regression random forests for binary risk estimation (as proposed by Malley et al. [2012]), random forests with probability estimation trees are evaluated. In addition, due to observed overfitting of random forests in the construction of GRS [Lau, 2020], a random forests variant that conducts a prior variable selection is also considered. Furthermore, ensemble logic regression using bagging is evaluated as well.

1.9 Detecting gene–environment interaction effects

As discussed in Section 1.2, environmental risk factors may also influence the manifestation of phenotypes. For example, the epidemiological SALIA cohort study (that was introduced in Section 1.3) investigates the role of air pollution in the development of chronic diseases. Often, genetic and environmental risk factors are analyzed separately, e.g., in GWAS that investigate the influence of genetic risk factors or in EWAS that investigate the influence of environmental risk factors. However, genetics and the environment might not only influence the development of phenotypes individually, but they might also interact with each other [Ottman, 1996].

These interactions are known as *GxE* (*gene–environment*) *interactions* and are defined as varying environmental effects on a considered phenotype for different genotypes. For example, an individual might be only susceptible to a pollutant if the individual carries a specific genetic makeup, as illustrated in Figure 2, where the environmental effect on the phenotype is only active (i.e., the slope is non-zero) if Genotypes II or III (but not Genotype I) are present. Vice versa, exposure to environmental risk factors might increase the disease risk effect of genotypes [Ottman, 1996]. If only Genotypes II and III are considered in Figure 2, there is no GxE interaction effect present, since, in this case, the environmental effects (i.e., the slopes) are equal and only the genetic main effects (i.e., the offsets) differ between these two genotypes.

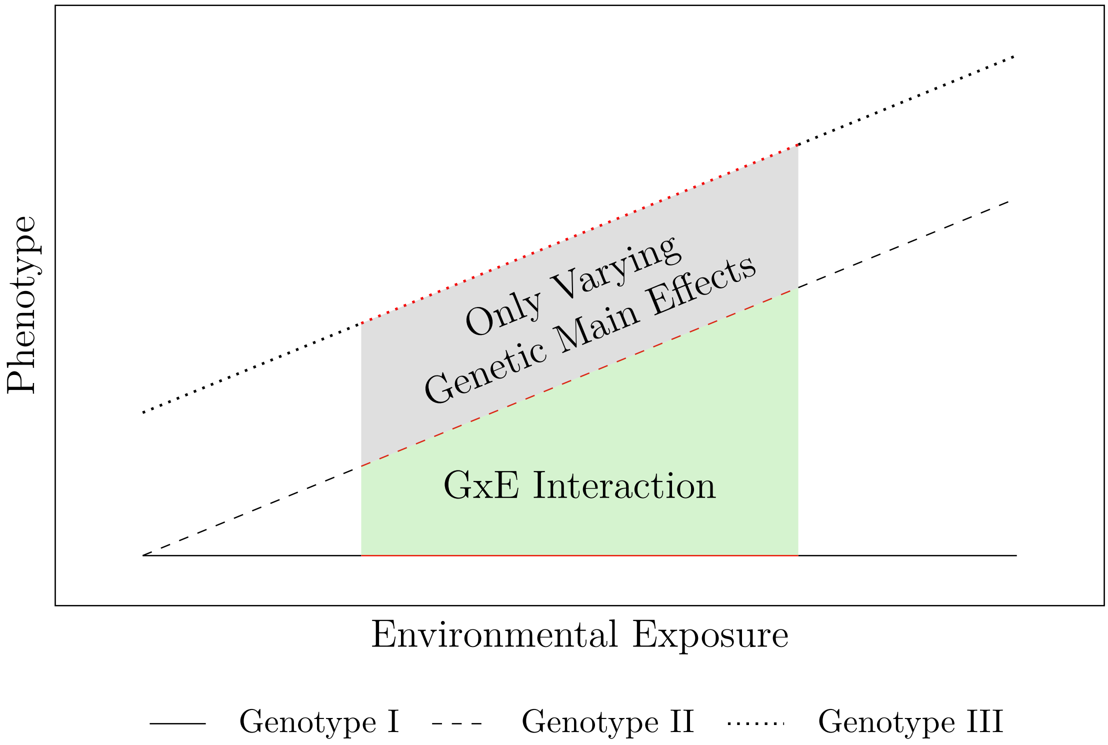

Figure 2: Environment–phenotype relationships for three different genotypes illustrating present (Genotype I versus Genotypes II/III) or absent (Genotype II versus Genotype III) gene–environment interaction effects

GxE interactions are commonly tested using GLMs

$$g(\mathbb{E}[Y \mid G, E, \boldsymbol{C}]) = \beta_0 + \beta_1 G + \beta_2 E + \beta_3 G \cdot E + \sum_{j=1}^{m} \gamma_j C_j, \tag{5}$$

where G is a genetic variable such as a SNP or a GRS, E is an environmental variable such as the exposure to an air pollutant, and $\boldsymbol{C} = \begin{pmatrix} C_1 & \ldots & C_m \end{pmatrix}^T$ are potential confounders such as BMI (body mass index), age, or sex that should be adjusted for (see Section 1.2) [Hüls et al., 2017a]. In this model, a GxE interaction is present if β_3 is unequal to zero. Therefore, to statistically assess whether there is a GxE interaction between G and E regarding Y, the statistical hypotheses $H_0 : \beta_3 = 0$ vs. $H_1 : \beta_3 \neq 0$ are tested, e.g., using a Wald test.

Usually, the genetic variable used in the GxE interaction testing model in Equation (5) is a single SNP. To test a whole genomic region for interaction with an environmental risk factor, this test is repeated for every considered SNP and the test results are gathered by performing a correction for multiple testing (usually employing a Bonferroni correction) [Majumdar et al., 2021] and choosing the minimum p-value for testing the hypothesis that the considered genomic region interacts in any way with E [Hüls et al., 2017a].

Since single SNPs are considered, correlations and interactions between SNPs are not taken into account. Moreover, a potentially conservative adjustment for multiple testing is performed. Hence, the statistical power, i.e., the probability of correctly rejecting the null hypothesis/correctly detecting a GxE interaction effect, of the single SNP test could be improved.

Therefore, a GRS that aggregates the genetic effects may be used in place of a single SNP as the genetic variable in Equation (5) for testing the presence of a GxE interaction [Hüls et al., 2017a]. However, if internal GRS shall be used, e.g., because no appropriate external weights are available, the available data set has to be split into training data for constructing the GRS and test data for testing the presence of a GxE interaction effect to avoid overfitting and to construct a statistically valid testing procedure. This need for splitting the data, again, reduces the statistical power due to less data being available for both steps.

Recently, many GxE interaction testing procedures have been proposed that do not require a data split [Gauderman et al., 2017]. These methods include two-stage methods that, similar to the single SNP test, first, perform individual tests for every considered SNP, and second, use these individual test results to test the global hypothesis that the considered genomic region interacts in any way with the considered environmental risk factor [Hsu et al., 2012, Gauderman et al., 2013, Lin et al., 2019].

Similar to the GRS-based GxE interaction test, SBERIA (set-based gene–environment interaction test) first constructs a weighted sum of SNPs that is used as the genetic variable G in Equation (5) in a second step to test the GxE interaction [Jiao et al., 2013]. However, to avoid overfitting and also the need for data splitting, the weights in this sum can only attain three different values, corresponding to no significant association, a significant positive association, or a significant negative association of the considered SNP. Therefore, these GxE interaction testing approaches perform limited modeling, not taking correlations or interactions between SNPs into account.

Another class of GxE interaction testing procedures are variance component tests that construct a GLMM (generalized linear mixed-effects model) including all considered SNPs, where the GxE interaction is interpreted as a random effect for which it is tested whether its variance is equal to zero [Lin et al., 2013, 2016, Su et al., 2016].

None of the discussed GxE interaction testing procedures could incorporate gene–gene interaction effects. This would, therefore, also reduce the statistical power for detecting GxE interactions if gene–gene interactions are present, as the genetic model would not be properly captured in this case.

To address the data splitting issue of the GRS-based GxE interaction test, we propose in Lau et al. [2023] (see Chapter 3) a GxE interaction test that is based on bagging for constructing the GRS model utilizing all available data and computing GRS predictions also for all observations using the OOB prediction mechanism. Furthermore, based on the observation that random forests can yield superior GRS compared to the elastic net (see Lau et al. [2022]/Chapter 2), we propose in Lau et al. [2023] using random forests as the GRS construction procedure to allow modeling flexibility also in the context of GxE interaction testing, as random forests can autonomously detect gene–gene interactions and naturally employs bagging. Moreover, random forests perform already relatively well with standard hyperparameter settings [Probst et al., 2019] so that hyperparameter optimization (and therefore, another data split) might not be mandatory.

The proposed GxE interaction testing procedure has been implemented in the R software package `GRSxE` that is publicly available on CRAN [Lau, 2023] (see Appendix A.1).

1.10 Interpretability of tree-based statistical learning methods

The in the previous Sections 1.7–1.9 considered tree-based statistical learning method random forests yields strongly predictive models. However, due to not considering one single decision tree but an ensemble of many decision trees, random forests are no longer as easily *interpretable* as, e.g., GLMs, where it can be directly seen and understood how exactly predictions are calculated.

Interpretable machine learning (also called explainable artificial intelligence) is the field that is concerned with making machine learning models understandable for humans [Holzinger et al., 2022]. Two of the most common concepts in interpretable machine learning are constructing inherently interpretable models such as GLMs or single decision trees and post-hoc interpretation of black-box models such as random forests or deep neural networks, e.g., by estimating the importance of predictors in the composition of the outcome [Holzinger et al., 2022, Bordt and von Luxburg, 2023].

Single logic regression models are more (inherently) interpretable than random forests, as usually not many logic trees are fitted in a model (the software implementation allows fitting a maximum of five logic trees [Kooperberg and Ruczinski, 2023]). However, if both Boolean conjunctions and disjunctions are allowed, the resulting Boolean expressions can be hard to interpret. Moreover, interactions are not directly revealed in such complex Boolean expressions.

Hence, to increase the interpretability of GRS without losing much predictive ability, an idea might be to construct single decision trees with an improved fitting procedure.

In Section 1.7.1, decision tree splits were introduced as triplets (X_j, A, A^c) for recursively splitting the predictor space on one predictor per split. However, for also taking interactions between predictors on split level into account, decision tree splits can be generalized to pairs (B, B^c), where $B \subsetneq \boldsymbol{\mathcal{X}}$ and $B^c = \boldsymbol{\mathcal{X}} \setminus B$ are complementary sets in the total p-dimensional predictor space. With this generalized split definition, decision tree splits can take multiple predictors at once into account. One established class of decision trees that construct these multivariate splits are oblique decision trees [Murthy et al., 1994] that split on linear decision rules

$$\left(\left\{\boldsymbol{x} \;\middle|\; \boldsymbol{x}^T\boldsymbol{a} < \delta\right\}, \left\{\boldsymbol{x} \;\middle|\; \boldsymbol{x}^T\boldsymbol{a} \geq \delta\right\}\right),$$

where $\boldsymbol{a} \in \mathbb{R}^p$ is a (possibly sparse) p-dimensional real vector that defines together with the decision threshold $\delta \in \mathbb{R}$ a hyperplane in $\mathbb{R}^p$.

These multivariate splits can describe interactions between predictors on split level and can lead to sparser decision trees [Hada and Carreira-Perpiñán, 2022]. However, the exact type of interaction cannot be directly inferred from the linear decision rules. One alternative might be to consider Boolean conjunctions of binary predictors (or, more generally, products of predictors) as splitting variables to directly reveal the interactions on split level.

In Figure 3**b**, such a decision tree that splits on the Boolean conjunction $X_1^c \wedge X_2$ is illustrated. This decision tree is—in comparison to the standard decision tree depicted in Figure 3**a** that utilizes univariate splits—sparser. Furthermore, the interaction effect can be directly read off from the decision tree that splits on a Boolean conjunction. Moreover, prediction value estimation becomes more robust, as more observations can be utilized. For example, the estimates for the prediction

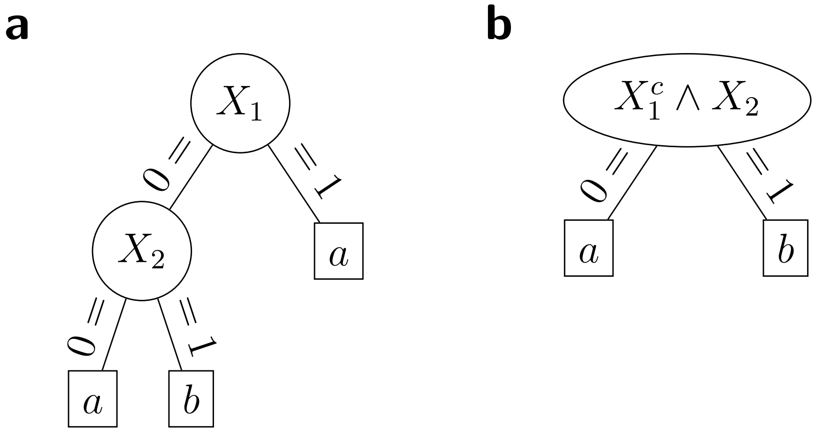

Figure 3: Decision trees that describe the same prediction function. In **a**, a standard decision tree using univariate splits is shown. In **b**, a decision tree that splits on the Boolean conjunction of two predictors is presented. Adapted from Lau et al. [2024].

value a in the standard decision tree from Figure 3**a** could be very different if few observations are available, as a is estimated twice in this tree using two independent samples, once for $(X_1 = 0) \wedge (X_2 = 0)$ and once for $X_1 = 1$. On the contrary, the decision tree splitting on a Boolean conjunction from Figure 3**b** estimates a only once for the complete sample with $X_1^c \wedge X_2 = 0$ which is equivalent to $(X_1 = 1) \vee (X_2 = 0)$.

Hence, to construct one interpretable and highly predictive GRS model, we develop in Lau et al. [2024] (see Chapter 4) a statistical learning method for constructing single decision trees that is tailored to binary predictors (such as SNPs divided into dominant and recessive modes of inheritance) and can identify splits on conjunctions of predictors. The proposed method is called *logicDT* (*logic decision trees*) and identifies influential predictors and interactions between predictors using an adaptive version of simulated annealing that—in contrast to logic regression—does not require a manual setup.

Decision tree procedures including logicDT recursively partition the predictor space which is especially useful for categorical variables such as SNPs that can attain only finitely many values, as, in this case, the true regression function $\mu(\boldsymbol{x}) = \mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]$ can be exactly represented by a decision tree. However, standard decision tree procedures can only approximate continuous relationships, that are induced by quantitative predictors such as environmental exposures, by step functions due to the prediction values being constant in decision tree leaves. Hence, an idea might be to fit regression models in the leaves to also properly incorporate continuous relationships [see, e.g., Zeileis et al., 2008]. In the setting of fitting parametric regression models such as GLMs, regression parameters instead of direct predictions have to be estimated in the leaves. The decision tree fitting algorithm, therefore, has to take this advanced modeling into account when identifying splits. This would allow fitting environment-phenotype models for different genotypes (i.e., based on splits considering single SNPs or gene–gene interactions), and thus, also modeling GxE interactions.

Therefore, we propose in Lau et al. [2024] fitting regression models based on quantitative covariates in the logicDT leaves to also properly model GxE interaction effects—as opposed to random forests that can only approximate continuous relationships using step functions and logic regression that cannot include interactions with continuous predictors. In logicDT, a splitting criterion based on likelihood-ratio tests is employed that compares the maximized leaf regression model likelihoods to decide if a node will be split and, if the node shall be split, based on which predictor/term the node will be split.

Similar to logic regression, in situations where the true underlying model is complex and many predictors influence the outcome, fitting a logicDT ensemble using bagging might further increase the predictive performance at the cost of no longer obtaining an inherently interpretable model. Hence, to retain some interpretability, the fitted logicDT ensemble model might be post-hoc explained.

For deriving which predictors influence the outcome in which magnitude in black-box models, *VIMs* (*variable importance measures*) can be employed [Breiman and Cutler, 2003, Hastie et al., 2009]. VIMs usually compare the full model containing all predictors to an informatively reduced model, where the considered predictor is no longer connected to the outcome, e.g., by randomly permuting the considered predictor or refitting the model without the considered predictor [Mentch and Hooker, 2016]. Bureau et al. [2005] proposed a joint VIM as follows that estimates the importance of multiple predictors at once by reducing the model also by multiple predictors at once.

Definition 5 (Variable importance measure)**.** Let $\epsilon(A)$ measure the error of a prediction model that (informatively) only uses the predictors in $A \subseteq \boldsymbol{X}$ (where $\boldsymbol{X} = \begin{pmatrix} X_1 & \dots & X_p \end{pmatrix}^T$ is interpreted as a set $\{X_1, \dots, X_p\}$). The (joint) *VIM* (*variable importance measure*) of $k \geq 1$ predictors $X_{j_1}, \dots, X_{j_k}$ is given by

$$\mathrm{VIM}(X_{j_1}, \dots, X_{j_k}) = \epsilon(\boldsymbol{X} \setminus \{X_{j_1}, \dots, X_{j_k}\}) - \epsilon(\boldsymbol{X}).$$

VIMs are typically computed on holdout/test data sets to avoid overfitting of the estimated importances. In the context of ensemble models fitted using bagging, the prediction errors may also be computed based on OOB predictions to obtain unbiased VIMs [Nicodemus et al., 2010, Janitza et al., 2013].

More recently, SHAP (Shapley additive explanation) values have been proposed for measuring the attribution of predictors in prediction models [Lundberg and Lee, 2017]. SHAP values are, as the name suggests, based on Shapley values from game theory that additively distribute the game's outcome among the participating players [Shapley, 1953]. In the case of SHAP values $\phi_j(f, \boldsymbol{x})$ $(j \in \{1, \ldots, p\})$, which are attributions of predictors X_j to the output $f(\boldsymbol{x})$ of a prediction function f, an additive decomposition of the prediction

$$f(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{X}}[f(\boldsymbol{X})] + \sum_{j=1}^{p} \phi_j(f, \boldsymbol{x})$$

is obtained [Lundberg and Lee, 2017, Loecher, 2023] that can be also interpreted as a GAM (generalized additive model), in which the individual attribution functions $\phi_j(f, \cdot)$ might depend on multiple predictors at once [Bordt and von Luxburg, 2023]. SHAP values have been extended to also measure the attribution of interactions between predictors using the Shapley interaction index [Fujimoto et al., 2006, Lundberg et al., 2020]. However, since SHAP values measure the attribution of predictors by comparing direct predictions of the considered model—and do not consider the model error (on independent test data) as conventional VIMs, predictor attributions measured using SHAP values might be high due to overfitting and not due to an actual influence on the outcome [Loecher, 2022]. Hence, SHAP values measure the attribution of predictors regarding model structure, whereas classic VIM approaches measure the importance of predictors regarding predictive performance/association with the outcome.

To quantify the effects of SNPs and certain gene–gene interactions on a considered phenotype and to obtain post-hoc interpretability of ensemble logicDT models, we also propose in Lau et al. [2024] a framework for measuring the influence of predictors and specific interactions of predictors on the outcome—as opposed to standard VIMs that either measure the influence of single predictors or joint influences, and therefore, do not measure the importance of isolated interaction effects—that can be used in conjunction with logicDT. The proposed interaction VIM is derived by recursively splitting joint effects into main and interaction effects and considers a general definition of interactions, i.e., if the considered predictors interact in any way. To also identify which specific Boolean conjunction is most likely associated with the considered interaction effect, all possible conjunctions are tested for their association with the outcome. As binary predictors can attain only two different values, the logic VIM is proposed that quickly estimates the informatively reduced model by considering all possible scenarios for predictors/terms for which the importance shall be estimated.

logicDT has been implemented in the R software package `logicDT` that is publicly available on CRAN [Lau, 2024] (see Appendix A.2).

1.11 Aims of this work

To summarize, this dissertation is concerned with GRS construction approaches that employ statistical learning. As discussed in the previous sections, there are several research gaps in this context that are addressed in this work, namely

1. incorporating gene–gene interaction effects in modeling GRS (Chapters 2–4),
2. investigating predictive performances/associations of the constructed GRS with the considered phenotype of tree-based procedures compared to standard GRS approaches in various situations (Chapter 2),
3. utilizing complex GRS models for statistically testing the presence of GxE interaction effects (Chapter 3),
4. identifying and properly modeling gene–gene and GxE interactions in one single model (Chapter 4),
5. constructing a highly predictive and highly interpretable GRS model (Chapter 4),
6. estimating the influence of SNPs and specific gene–gene interaction effects on the considered phenotype (Chapter 4).

Contributing to filling these research gaps could improve GRS construction in practice so that GRS estimation in precision medicine could become more accurate. Moreover, improving the statistical power of GxE interaction testing procedures and improving the interpretability of constructed GRS models could lead to revealing so-far hidden biological mechanisms in the development of complex diseases.

In Chapter 2, the first publication is presented that addresses Research Gaps 1 and 2. Chapter 3 consists of the second publication that addresses Research Gaps 1 and 3. The third publication is presented in Chapter 4 and addresses Research Gaps 1 and 4–6. Chapter 5 concludes this dissertation and provides discussions on potential future research.

Chapter 2

Evaluation of tree-based statistical learning methods for constructing genetic risk scores

In the following, the first manuscript [Lau et al., 2022], which was published in the journal BMC Bioinformatics, is presented and addresses Research Gaps 1 and 2.

The first simulation scenario considered in this manuscript has been also evaluated in a slightly modified version in the master's thesis *Evaluation of Tree-Based Classification and Regression Methods for Constructing Genetic Risk Scores* [Lau, 2020]. However, in this paper, different variants of the tree-based GRS construction approaches are considered than in the master's thesis (see also Section 1.8).

Evaluation of tree-based statistical learning methods for constructing genetic risk scores

Michael Lau, Claudia Wigmann, Sara Kress, Tamara Schikowski, and Holger Schwender

Authorship:	First author
Contributed part:	80%
Contribution:	Adapting the methods for GRS construction Designing the simulation study Conceiving the real data analysis Conducting the statistical data analyses Preparing figures and tables Interpretation of results Writing the paper
Journal:	BMC Bioinformatics
Impact factor:	3.025 (2022)
Date of publication:	21 March 2022
DOI:	10.1186/s12859-022-04634-w

BMC Bioinformatics

RESEARCH Open Access

Evaluation of tree-based statistical learning methods for constructing genetic risk scores

Michael Lau[1,2*], Claudia Wigmann[2], Sara Kress[2], Tamara Schikowski[2] and Holger Schwender[1]

*Correspondence:
michael.lau@hhu.de
[1] Mathematical Institute, Heinrich Heine University, Düsseldorf, Germany
Full list of author information is available at the end of the article

Abstract

Background: Genetic risk scores (GRS) summarize genetic features such as single nucleotide polymorphisms (SNPs) in a single statistic with respect to a given trait. So far, GRS are typically built using generalized linear models or regularized extensions. However, these linear methods are usually not able to incorporate gene-gene interactions or non-linear SNP-response relationships. Tree-based statistical learning methods such as random forests and logic regression may be an alternative to such regularized-regression-based methods and are investigated in this article. Moreover, we consider modifications of random forests and logic regression for the construction of GRS.

Results: In an extensive simulation study and an application to a real data set from a German cohort study, we show that both tree-based approaches can outperform elastic net when constructing GRS for binary traits. Especially a modification of logic regression called logic bagging could induce comparatively high predictive power as measured by the area under the curve and the statistical power. Even when considering no epistatic interaction effects but only marginal genetic effects, the regularized regression method lead in most cases to inferior results.

Conclusions: When constructing GRS, we recommend taking random forests and logic bagging into account, in particular, if it can be assumed that possibly unknown epistasis between SNPs is present. To develop the best possible prediction models, extensive joint hyperparameter optimizations should be conducted.

Keywords: Polygenic risk scores, Epistasis, Statistical learning, Random forests, Variable selection, Logic regression, Bagging, Elastic net, Simulation study

Background

The development of complex diseases depends on many factors such as genetic mutations, the lifestyle, or environmental factors. Investigating the effects of genetic variants across the human genome in genome-wide association studies (GWAS) has already revealed relevant risk base-pair alterations [1]. Single nucleotide polymorphisms (SNPs) may have only a very small effect on the investigated disease. However, when considered jointly, SNPs might be highly relevant [2, 3]. This behavior can be due to many independent SNPs exhibiting minor individual effects, or it can be caused by interactions of genetic variants, i.e., epistasis.

In consequence, summarizing relevant genetic effects in an individual while sufficiently predicting the risk for a certain disease, potentially jointly with non-genetic covariables, would be highly desirable. This would, on the one hand, allow to uncover underlying mechanisms related to this specific disease. On the other hand, accurately predicting the risk of disease for an individual could have a high impact on personalized medicine due to potentially being able to reduce the personal risk by taking specialized preventive measures if an individual has a high genetic risk for a certain disease [4, 5].

One promising approach for the assessment of an individual's risk is the development of genetic risk scores (GRS). For the construction of GRS, one typically selects a subset of relevant SNPs from a biological pathway or a gene and calculates a weighted sum of the selected genetic variants.

Genome-wide approaches with a selection of genetic variants from across the whole genome resulting from prior knowledge are also possible for building GRS [6, 7]. However, such selections typically depend on large-scale association studies in which single SNPs were tested individually with regard to the phenotype. Thus, interacting variants which do not exhibit substantial marginal effects might be left out although SNP level interactions might contribute to disease risk [8, 9]. In this context, an alternative to conventional GWAS for identifying disease-related SNPs might be genome-wide association interaction studies (GWAIS) [9].

The standard procedure for the computation of the GRS is the usage of external weights [10, 11], ideally determined from independent association studies such as GWAS or GWAIS. However, there might be no appropriate association study for the regarded outcome or population available such that suitable weights have to be gathered in a different way.

Internal GRS weights can be estimated by regarding the problem of constructing GRS as a supervised statistical learning problem, where the response would be the disease status or a quantitative biological variable such as the glucose level. In this case, the predictors are genetic variants of the specific pathway or gene, where SNPs are usually coded by the number of minor alleles for this individual. The estimation of proper weights or fitted models which generalize well, i.e., which represent the whole population reasonably well and not only the available sample, requires the partitioning of the whole data set into training and test data sets. Dudbridge [3] and Hüls et al. [11] found in their studies that a random close to one-half split generalizes well. Sufficient samples are necessary in the test data set for evaluating the association of the GRS with the response which especially holds true for gene-environment interaction (GxE) studies in which more parameters are to be estimated. A GxE interaction is present if, for different genotypes, different disease susceptibilities to an environmental factor are underlying, e.g., if an individual has a high genetic risk for a certain disease which is enabled by an environmental factor [12].

So far mainly linear methods such as generalized linear models (GLM) or regularization methods based on GLMs, such as the lasso [13] or one of its generalizations, the elastic net [14], have been used in the construction of GRS [11, 15, 16]. The elastic net offers the advantage of properly handling highly correlated predictors, e.g., SNPs in linkage disequilibrium (LD), by employing an L_2 regularization while performing a variable selection due to the L_1 regularization. Nonetheless, these regularized linear regression methods cannot directly take interactions between predictors into account (unless specific interaction terms were specified prior to applying them) and the assumption of an additive relationship between the response and the input variables has to be fulfilled. Therefore, the usage of algorithms which are able to develop more general models and which in fact can find and take interesting interactions into account might be preferable.

The tree-based statistical learning method random forests [17] is well-known and widely used among a variety of use cases [e.g., [18–20]]. It builds several individual classification or regression trees (CART) [21], which are fitted by a non-linear recursive partitioning algorithm, and combines them to one strong ensemble. For a low to moderate amount of SNPs (< 100), it has been shown that the classic random forests algorithm is able to properly uncover SNP interactions even when the corresponding marginal effects are negligible [22].

Another tree-based non-linear statistical learning procedure is logic regression [23] which mainly considers binary predictors. It searches for Boolean expressions of the input variables and combines multiple expressions in a GLM and already has been used in applications to SNP data [24–26]. Both tree-based methods are theoretically able to cover each possible prediction scenario for categorical input data. However, their model fitting techniques are highly different.

To the best of our knowledge, it has barely been investigated yet whether the aforementioned statistical learning algorithms can be used as alternative procedures to conventional GRS construction approaches. For random forests, some publications suggest that the ensemble method is able to outperform conventional linear methods such as logistic regression, odds ratio scores or the lasso [27, 28]. However, more recent studies which considered genome-wide risk scores, i.e., GRS constructed using SNPs from all over the genome and not just single genes or pathways, were not able to verify that random forests should be used over linear approaches [29, 30]. In the context of disease risk prediction, e.g., Yoo et al. [31] regarded random forests, logic regression, and logistic regression without penalization in one simple gene-gene interaction simulation study and additionally in a real data application. In their analyses, the tree-based algorithms could induce higher predictive performances than logistic regression. Nonetheless, multi-faceted analyses taking different realistic data scenarios into account are necessary in order to draw meaningful conclusions about the appropriateness of the tree-based methods for the construction of GRS.

The classic random forests and logic regression algorithms have some shortcomings. In particular, random forests can severely overfit the data [32] and logic regression can lead to highly variant models [24]. Thus, we additionally considered modifications of the classic algorithms to overcome these drawbacks.

In this article, we, therefore, evaluate random forests, logic regression, and extensions of these methods in an extensive simulation study and an application to a real data set from a German cohort study for the construction of GRS and compare the results to the elastic net.

Methods

Construction of genetic risk scores

Let $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ be a training data set with N observations and binary outcomes $y_i \in \{0, 1\}$. Further assume that each input vector $\boldsymbol{x}_i$ is a collection of p biallelic SNPs, i.e., taking values in the p-dimensional space $\{0, 1, 2\}^p$, where 0 codes the homozygous reference, 1 the heterozygous variant, and 2 the homozygous variant. Then the problem of constructing a GRS model consists of fitting a proper function

$$\mathrm{GRS} : \{0, 1, 2\}^p \to [0, 1].$$

The target space is equal to the probability scale [0, 1], since $\mathrm{GRS}(\boldsymbol{x})$ should be an estimate of $\mathbb{P}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$, i.e., the probability of being a case given some SNPs $\boldsymbol{x}$. This fitting procedure is conducted on a designated training data set. Independent observations from a test data set $\mathcal{D}_{\text{test}}$ are used to evaluate the GRS, i.e., $\mathrm{GRS}(\boldsymbol{x})$ for $(\boldsymbol{x}, \cdot) \in \mathcal{D}_{\text{test}}$.

An overview of the workflow for fitting and evaluating GRS models using the statistical learning approach is given in Fig. 1.

Random forests

In random forests, multiple classification or regression trees (CART) [21] with injected randomness are built to form one strong ensemble. From a graph-theoretical point of view, decision trees are usually binary trees in which each inner knot represents a split based on a predictor and each leaf (terminal node) describes a prediction

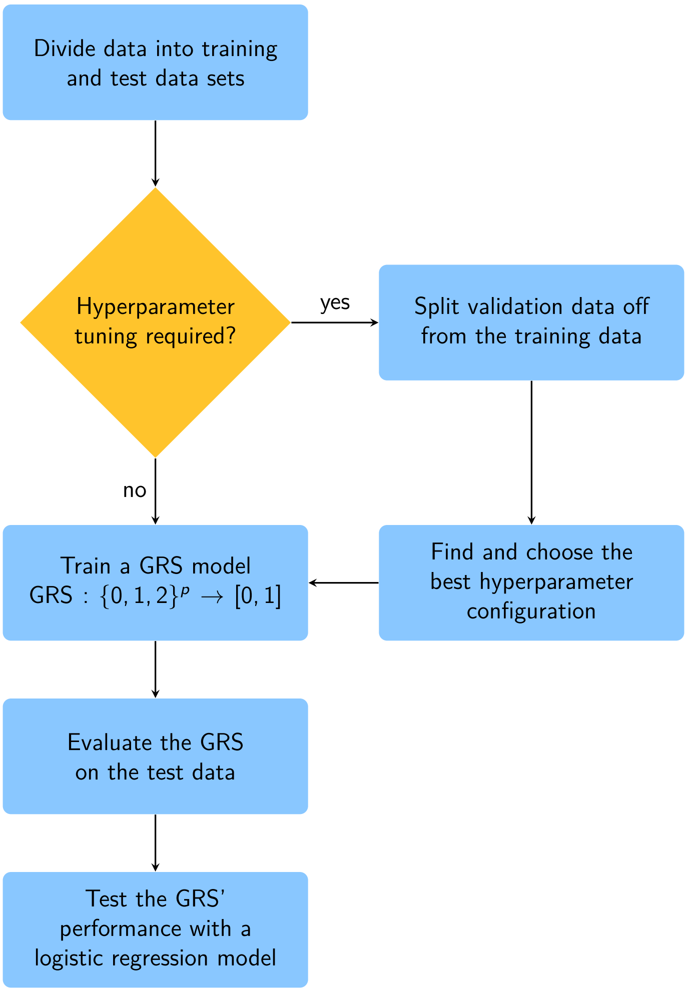
Divide data into training and test data sets
Hyperparameter tuning required?
yes
Split validation data off from the training data
no
Train a GRS model GRS : $\{0, 1, 2\}^p \to [0, 1]$
Find and choose the best hyperparameter configuration
Evaluate the GRS on the test data
Test the GRS' performance with a logistic regression model

Fig. 1 Workflow of constructing and evaluating genetic risk scores

scenario. Figure 2a illustrates an exemplary classification tree with four disjoint prediction scenarios. New predictions start at the root node and follow the respective edge until a leaf is reached.

Decision trees are induced by a recursive greedy splitting algorithm which searches at each inner node for the best possible split with respect to an impurity measure. The impurity measure is a quantifier for the homogeneity of respective nodes. For binary classification trees, the Gini impurity

$$i(t) = 2 \cdot P(Y = 1 \mid \boldsymbol{X} \in t)P(Y = 0 \mid \boldsymbol{X} \in t)$$

for empirical probabilities $P(Y = c \mid \boldsymbol{X} \in t)$ that the response Y is equal to class c given that the input vector $\boldsymbol{X}$ falls into the regarded node t is usually chosen.

The tree induction procedure can be locally terminated by stopping criteria. When a node should not be split, it is declared as a leaf and has to receive a prediction value. For classification trees, this is usually the class with the highest empirical probability in the regarded branch.

However, single decision trees suffer from the instability problem which states that a small noise-like modification of the training data set may lead to a disproportional modification of the fitted tree. This issue is mainly caused by the greedy fashion of choosing splits [33].

Random forests tries to address this issue. The algorithm employs bagging [34] which draws a bootstrap sample of the available data for each individual tree as its training data set. The tree fitting procedure is further randomized by adjusting the splitting algorithm to choose $mtry \leq p$ predictors from the total set of input variables at every inner node which qualify for the best split. $mtry$ is a hyperparameter usually chosen as $\sqrt{p}$ or $p/3$ which should be properly tuned in certain applications. Based on these randomizations, the resulting model averages the individual trees, i.e., for classification trees, the class which is classified most often will be chosen as the prediction.

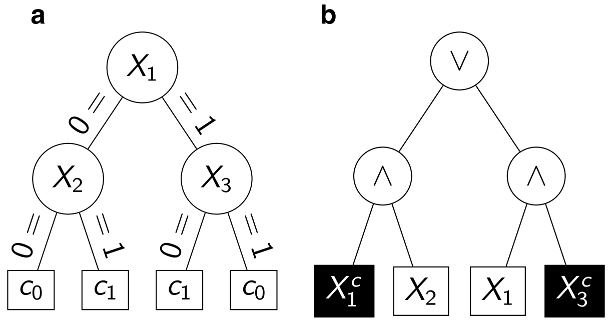

Fig. 2 Exemplary tree models for three binary input variables X_1, X_2 and X_3 predicting two different classes c_0 and c_1. In **a**, a classification tree is shown. **b** depicts a logic tree describing the Boolean expression $(X_1^c \wedge X_2) \vee (X_1 \wedge X_3^c)$. Here, a true Boolean expression is identified as class c_1 and c_0 otherwise. Negated input variables/leaves are marked by white letters on a black background. Both trees are equivalent, i.e., they perform the same predictions for each predictor setting

Random forests for constructing genetic risk scores

If one is aiming at constructing GRS for binary traits, one has to keep in mind that probability estimates for showing the regarded feature are needed instead of class estimates. Random forests based on classification trees can be used for probability estimation by averaging the number of trees which voted for class 1 [35]. However, if we, e.g., assume that the true risk for being a case would be equal to 80% and that all classification trees properly recognize this fact and, therefore, predict class 1 for this particular setting, the forest risk estimate would be given by 100%. Thus, for this reason, we consider probability estimation trees [36] which hold risk estimates in their leaves in contrast to classifications. These estimates are usually chosen as the empirical branch probabilities from which classification trees also draw their estimates. Random forests based on probability estimation trees average the probability estimates of the individual trees similar to regression trees.

If SNP variables coded as 0, 1, or 2 are interpreted as quantitative variables, decision trees and random forests are able to split with respect to $(\{0\}, \{1, 2\})$ or $(\{0, 1\}, \{2\})$, thus, considering both dominant and recessive modes of inheritance. Therefore, SNPs are directly used as input variables when employing random forests.

Random forests VIM

One issue that arose when fitting the first GRS models with random forests in our initial experiments was a substantial overfitting which could be observed by comparing the test and training data errors. Therefore, performing an appropriate variable selection prior to fitting the final random forests models might reduce the overfitting and lead to better results for noise-intensive data. Kursa and Rudnicki [37] proposed an iterative variable selection approach which relies on variable importance measures (VIM) and which they called Boruta. The permutation VIM can be calculated using the out-of-bag observations for each tree, thus, avoiding an overfitting of the VIM itself. In each iteration, the Boruta approach adds for each predictor variable a shadow variable with the same values but randomly permutes them to destroy a potential predictor-response relationship for this variable. Next, a random forest on this extended set of input variables is fitted and the evaluated VIMs for these shadow variables are used to approximate the distribution of VIMs for non-influential input variables. The computed VIMs of the original variables are then compared to the VIMs of the shadow variables in statistical tests for importance. In particular, the maximum observed importance of all shadow variables is used to decide whether an original variable is temporarily classified as important. More specifically, if a variable yields an importance higher than the maximum observed importance among all shadow variables, it will be temporarily marked as important. Several iterations of creating shadow variables, fitting random forests, and computing VIMs are used to perform binomial tests, which regard how often the variable was temporarily marked as important, testing the alternative of greater or smaller VIM realizations, i.e., important or unimportant variables. More precisely, these binomial tests are based on the null hypothesis that the probability of the regarded input variable yielding a higher VIM than the maximum VIM of all shadow variables is equal to 0.5. The significance threshold of the binomial tests is set to 1%, which is also the recommended threshold by the authors of the Boruta approach. Compared to other random-forest-based variable selection methods such as the Vita algorithm proposed by Janitza et al. [38] which relies on negative VIM values, the Boruta approach does not require a vast amount of (noninfluential) input variables.

As an alternative procedure, we also tried the variable selection method by Altmann et al. [39], which relies on random permutations of the response variable. However, in our experiments, the Boruta approach yielded more stable results in general. In particular, even when considering different significance thresholds for the approach by Altmann et al. [39], the Boruta procedure still could induce more stable variable selections, i.e., leading to variable selections that did not severely differ between independent replicates. This observation is in line with the analyses by Degenhardt et al. [40] who provide an in-depth comparison of various random forests variable selection methods.

Hence, we fitted ordinary random forests with probability estimation trees and random forests based on the Boruta variable selection which we call random forests VIM in the following. For random forests, we used the `R` package `ranger` [41]. For random forests VIM, the `R` package `Boruta` [37], that also relies on the `ranger` package, was used.

Logic regression

Logic regression [23] is a tree-based statistical learning algorithm which is specifically tailored to binary input variables. It searches for ideal Boolean expressions of those and works with binary tree representations of Boolean expressions, logic trees. Logic trees hold the Boolean operators $\wedge$ (AND) or $\vee$ (OR) in their inner nodes and contain predictor variables or their negations (indicated through ^{c}) in their terminal nodes. Figure 2b depicts an exemplary logic tree which is equivalent to the exemplary classification tree from Fig. 2a, i.e., both trees perform the same predictions for each realization of the three input variables. The interpretation as a Boolean expression is obtained recursively by combining expressions in a bottom-up fashion, yielding $(X_1^c \wedge X_2) \vee (X_1 \wedge X_3^c)$ for the logic tree from Fig. 2b.

Logic trees themselves can only be used for binary classification tasks, since they represent logic expressions so that their output is also either 0 or 1. To generalize their usage for, e.g., risk prediction, Ruczinski et al. [23] proposed using logic trees $L_1, \ldots, L_M$ as predictors in a GLM

$$g(\mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]) = \beta_0 + \beta_1 L_1(\boldsymbol{x}) + \ldots + \beta_M L_M(\boldsymbol{x})$$

considering an appropriate link function g such as the logit function $\operatorname{logit}(p) = \log(p/(1-p))$ for a binary response.

The total model fitting procedure consists of finding the most appropriate logic tree(s). In practice, for each model, a set of neighbor states is defined by simple adjustments of the current model. The moves used in logic regression consist of exchanging variables and operators, adding or removing branches, splitting or removing variables, and adding or removing trees. This set of moves ensures that from every state, every other possible state can be reached in a final number of steps. For more details, see [23].

Based upon this methodology, two model search algorithms are used in practice:

- a greedy search which evaluates each neighbor of a given state and moves to the best one
- simulated annealing [42], a stochastic search algorithm which only considers one random neighbor per iteration and can also move to worse states to prevent being stuck in a local minimum.

Model ranking is performed using a score function which is chosen to be the deviance for the logistic model. The model which yields the best score among all models visited in the search is chosen as the resulting model. Irrespective of using the greedy approach or simulated annealing, one should configure the model size hyperparameters, i.e., the total number of trees and the total number of leaves, to obtain the best fit on the entire population. For fitting conventional logic regression models, we used the `R` package `LogicReg` [43] and used simulated annealing as the search procedure.

Logic regression for constructing genetic risk scores

SNP variables coded as 0, 1, or 2 can be biologically meaningful divided into two binary variables, in $\text{SNP}_D = \mathbb{1}(\text{SNP} \neq 0)$, coding for a dominant effect, and in $\text{SNP}_R = \mathbb{1}(\text{SNP} = 2)$, coding for a recessive effect. With these two binary variables, interactions can be properly expressed. For example, consider a scenario where two SNPs influence the disease risk in such a way that the risk is significantly increased if and only if for both SNPs their respective minor allele occurs at least once. With Boolean logic, this can be expressed as $\text{SNP}_{1,D} \wedge \text{SNP}_{2,D}$. It might also be possible that two risk-increasing SNPs with a dominant mode of inheritance can only elevate the disease risk once, i.e., if both statuses occur, the risk is not increased beyond the first elevation. This scenario can also be expressed with Boolean logic as $\text{SNP}_{1,D} \vee \text{SNP}_{2,D}$. Furthermore, SNPs in high linkage disequilibrium (LD) that are, therefore, highly correlated can also be properly addressed with the logical OR. One LD block might then be expressed as a chain of OR-concatenated SNPs, a disjunction. Thus, for the construction of GRS with logic regression, each SNP is divided into two binary variables prior to applying the procedure.

Logic bagging

As an alternative to an exhaustive search with simulated annealing, we also considered applying bagging [34] to logic regression models fitted with a greedy search. We call this approach logic bagging. In contrast to conventional logic regression, logic bagging fits ensembles of individual logic regression models and, similar to random forests, predictions are made using the average of the predictions of the individual logic regression models. This approach is still computationally expensive when using an adequate amount of bagging iterations (e.g., 500) but reduces the variance and does not require the tuning of a cooling schedule. Logic bagging is implemented in the `R` package `logicFS` [44]. For fitting logic bagging models, the greedy search is employed mainly due to computational reasons. In particular, in Additional file 1: Fig. S1, the model fitting times are depicted. For example, for fitting and evaluating a single logic bagging model consisting of 500 logic regression models fitted via simulated annealing, it would take about

$500 \cdot 28.82s \approx 4h$ using the mean model fitting and evaluation time of $28.82s$ for logic regression.

Elastic net

The elastic net [14] is a regularized linear regression model which combines

- the lasso (least absolute shrinkage and selection operator) [13], i.e., L_1 regularized regression that reduces the estimate of the regression coefficients of non-influential predictors to zero, therefore, excluding non-informative input variables,
- and ridge regression [45], i.e., L_2 regularized regression for properly handling highly correlated predictors by assigning similar weights to such predictors.

Elastic net, hence, uses a penalty term given by

$$R_\alpha(\boldsymbol{\beta}) := \frac{1}{2}(1-\alpha)||\boldsymbol{\beta}||_2^2 + \alpha||\boldsymbol{\beta}||_1$$

for the regression coefficients $\boldsymbol{\beta} = \begin{pmatrix}\beta_1 & \ldots & \beta_p\end{pmatrix}^T$ in the fitting procedure solving the optimization problem

$$\min_{\beta_0,\boldsymbol{\beta}} \left\{ -\frac{1}{N}\ell(\beta_0,\boldsymbol{\beta}) + \lambda R_\alpha(\boldsymbol{\beta}) \right\} \tag{1}$$

for the log-likelihood function ℓ. In this article, binary outcomes are considered. Thus, the logistic regression approach for elastic net was employed.

Here, $\lambda \geq 0$ determines the strength of the regularization, i.e., for larger values of λ, the penalty $\lambda R_\alpha(\boldsymbol{\beta})$ increases, thus, favoring coefficient vectors with smaller norms, i.e., more loosely fitting models. The parameter $\alpha \in [0,1]$ configures the balance between the lasso and ridge regression, i.e., for $\alpha = 0$, one would perform ordinary ridge regression and for $\alpha = 1$, one would apply the lasso. Therefore, these two hyperparameters have to be tuned properly.

In practice, the model coefficients are estimated by employing coordinate descent as optimization algorithm to solve the minimization problem (1) and taking advantage of the fact that similar values of λ lead to similar model coefficients for a fast fitting of different λ settings [46]. We used the `R` package `glmnet` [46] with cross-validation for fitting elastic net models.

The common procedure when constructing GRS with regularized regression procedures such as elastic net is to use the $\{0,1,2\}$ coding for each SNP in the model [11, 16]. We, therefore, follow in our comparison this standard procedure and use the $\{0,1,2\}$ coding in the elastic net.

If interaction effects between SNPs should be included in the elastic net model, they have to be explicitly specified prior to fitting the model. However, in practice, it is usually unknown which loci might interact. Including all possible interactions between SNPs becomes rapidly infeasible, as for a moderate amount of SNPs, the number of possible interaction terms might already be too high. For example, for 50 SNPs, there exist more than 10^{15} interaction terms. The standard procedure for constructing GRS with linear methods such as the elastic net is to only consider marginal genetic effects [16]. Thus, we follow in our evaluations this common procedure and do not include interactions in the elastic net models.

Simulation studies

The tree-based statistical learning methods random forests, random forests VIM, logic regression, and logic bagging were evaluated and compared to the elastic net in a simulation study considering three scenarios with several different settings. All SNPs were drawn independently resembling LD-based pruned or clumped SNPs. All simulations and analyses were performed with `R` version 4.0.3 [47]. Data sets for all simulation scenarios were generated using the `R` function `simulateSNPglm` from the `scrime` package [48].

General workflow

The general workflow for generating the data sets for the simulation study is given as follows for each of the simulation settings, which are described in detail afterwards.

1. Choose the fixed data parameters, i.e., the odds ratios, number of SNPs, sample size and simulation design.
2. For each SNP, draw a random minor allele frequency (MAF).
3. Randomly generate the genotypes based on the MAFs.
4. If continuous covariables are to be included, randomly generate the data for these variables.
5. Randomly generate the outcome according to the linear predictor.
6. Evaluate the fraction of cases in the generated outcome and tune the prevalence such that this fraction becomes approximately balanced, i.e., yielding $\sim$ 50% cases. This involves going back to step 5.
7. Create 100 independent data sets for a certain setting using the steps 2–5 for each repetition.

Simulation setups

Marginal genetic effects

In a first step, we focused on main effects, which represents the ideal case for the elastic net, since no interactions are considered here and the individual effects behave additively with each other. Similar to Hüls et al. [49], we considered six SNPs influencing the value of the outcome, where we simulated a dominant effect for each of these SNPs. Thus, we first considered data sets generated from a logistic regression model

$$\operatorname{logit}(\mathbb{P}(Y=1)) = \beta_0 + \sum_{i=1}^{6} \beta_i \cdot \mathbb{1}(\mathrm{SNP}_i \neq 0) = \beta_0 + \sum_{i=1}^{6} \beta_i \cdot \mathrm{SNP}_{i,D}. \tag{2}$$

In order to draw conclusions for different realistic scenarios, we varied three parameters:

- the effect size, i.e., the odds ratio, of each influential SNP which can be configured by specifying $\exp(\beta_i)$ [50],

Table 1 Parameter settings for the first simulation scenario resulting in 27 settings in total

Parameter	Considered realizations
Odds ratio	1.2, 1.5, 1.8
Amount of noise SNPs	4, 14, 44
Sample size	500, 1000, 2000
Prevalence	Resulting in balanced data sets
MAF	Randomly chosen from [0.15, 0.45]
Repetitions	100

- the intensity of statistical noise which we adjusted by adding non-influential SNPs to each data set,
- and the sample size of each data set.

To achieve nearly case-control study-like designs, we configured the prevalence, i.e., $(1 + \exp(-\beta_0))^{-1}$ [50], to result in nearly balanced data sets for each regarded odds ratio. The MAF was drawn randomly for each SNP and for each data set from the interval [0.15, 0.45] similar to Pan et al. [51]. For each scenario, we generated 100 independent data sets, i.e., we performed 100 replications. Table 1 lists the regarded settings for the aforementioned simulation parameters.

Dominant interactions of SNPs

In a second simulation scenario, we additionally considered a gene-gene interaction, i.e., an interaction between SNPs. More specifically, we here always considered three SNPs with low main effects, i.e., odds ratios of 1.2 and a dominant mode of inheritance, since we focused on marginal effects in the first scenario. Additionally, we included an interaction term between two SNPs whose odds ratio was varied. Similar to the first scenario, we also varied the amount of statistical noise, i.e., the number of SNPs for which no effect on the outcome is intended. Furthermore, we considered three sub designs that determine which SNPs interact. The data was generated following models such as

$$\operatorname{logit}(\mathbb{P}(Y = 1)) = \beta_0 + \sum_{i=1}^{3} \beta_i \cdot \mathrm{SNP}_{i,D} + \beta_4 \cdot \mathrm{SNP}_{j,D} \cdot \mathrm{SNP}_{k,D}. \tag{3}$$

The indices $(j, k) \in \{(1, 2), (1, 4), (4, 5)\}$ determine whether both interacting SNPs also do have marginal effects, only one of them exhibits a main effect, or if they only are influential when considered jointly. The prevalence was again configured by β_0 to approximately achieve case-control-balanced study designs. The MAF was randomly chosen in the interval [0.15, 0.45] and the sample size was fixed to 2000 observations per data set, since we only considered weak marginal effects. 100 independent data sets for each setting were analyzed using a cyclic scheme such as in the first simulation scenario. The study parameters for the second simulation scenario are summarized in Table 2.

Gene-environment interactions

In the final simulation scenario, we added two correlated continuous variables to the true underlying model from which one forms a GxE interaction with a SNP. One of

Table 2 Study parameters for the second simulation scenario resulting in 45 settings in total

Parameter	Considered realizations
Odds ratio of gene-gene interaction	1.2, 1.5, 1.8, 2.1, 2.4
Amount of noise SNPs	5, 15, 45
Interacting SNPs (j, k)	(1, 2), (1, 4), (4, 5)
Sample size	2000
Prevalence	Resulting in balanced data sets
MAF	Randomly chosen from [0.15, 0.45]
Repetitions	100

these two variables exhibits a marginal effect on the outcome, while the second variable only influences the outcome if a certain risk allele occurs at least once. The data for this scenario was generated considering the model

$$\begin{aligned}\operatorname{logit}(\mathbb{P}(Y=1)) = \beta_0 + \sum_{i=1}^{3} \beta_i \cdot \mathrm{SNP}_{i,D} + \beta_4 \cdot \mathrm{SNP}_{1,D} \cdot \mathrm{SNP}_{4,D} \\ + \beta_5 \cdot E_1 + \beta_6 \cdot E_2 \cdot \mathrm{SNP}_{j,D}.\end{aligned} \tag{4}$$

Similar to the gene-gene interaction simulation scenario, the effects for the first three SNPs were fixed to odds ratio of 1.2, 1.5, and 1.8, respectively. The interaction between SNP_1 and SNP_4 received a fixed odds ratio of 1.8, since in this analysis, the focus lies on the GxE interaction. The index $j \in \{2, 5\}$ determines whether the SNP in the GxE interaction also exhibits a moderate marginal effect or if this SNP only influences the outcome in interaction with the continuous variable E_2. The odds ratios of the terms involving the continuous variables E_1 or E_2 were specified per IQR (interquartile range) of the respective environmental variable as it is regularly done when performing analyses of GxE interactions [11]. For the continuous variable E_1, the (marginal) odds ratio was fixed to 1.2 per IQR. The odds ratio of the GxE interaction between SNP_j and E_2 was varied between 1.2 and 2.4. The continuous variables were generated following a multivariate normal distribution, i.e.,

$$\begin{pmatrix} E_1 \\ E_2 \end{pmatrix} \sim \mathcal{N}_2\left(\begin{pmatrix} \mu \\ \mu \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right).$$

In particular, the mean μ was set to 20, the variance σ^2 was chosen as 10 and the correlation ρ between these two variables was chosen as either 0.5 or 0.9, resembling moderately and highly correlated variables, respectively. The prevalence was again configured by β_0 to approximately achieve case-control-balanced study designs. The MAF was randomly chosen in the interval [0.15, 0.45] and the sample size was fixed to 2000 observations per data set as in the previous simulation scenario. The number of additional noise SNPs was fixed to 45. 100 independent data sets for each setting were analyzed. The study parameters for the third simulation scenario are summarized in Table 3. In GxE interaction studies, GRS are usually constructed only using the available genetic data [11]. Thus, we constructed the GRS without utilizing the environmental variables.

Table 3 Study parameters for the third simulation scenario resulting in 20 settings in total

Parameter	Considered realizations
Odds ratio of GxE interaction	1.2, 1.5, 1.8, 2.1, 2.4
Amount of noise SNPs	45
Interacting GxE SNP j	2, 5
Correlation between E_1 and E_2	0.5, 0.9
Sample size	2000
Prevalence	Resulting in balanced data sets
MAF	Randomly chosen from [0.15, 0.45]
Repetitions	100

Analysis of association and predictive strength

To evaluate and compare the different statistical learning methods in their ability to construct GRS, a cyclic training-validation-test data set scheme was considered. In the i-th repetition of this cyclic scheme, the i-th data set $\mathcal{D}_i$, $i \in \{1, \dots, 100\}$, was used to train the GRS with the different statistical learning methods. For the evaluation of the performance of these methods, the succeeding data set $\mathcal{D}_{i+1}$ if $i \neq 100$ and $\mathcal{D}_1$ otherwise was chosen to be the independent test data set. For tuning the hyperparameters (see "Section Hyperparameter optimization"), we chose the preceding data set, i.e., $\mathcal{D}_{i-1}$ if $i \neq 1$ and $\mathcal{D}_{100}$ otherwise as validation data.

Since all data sets were generated independently, the cyclic scheme is equivalent to a conventional training-validation-test data set approach in which each of the 100 data sets is once used as training set, once as test set, and once as validation set in a cyclic manner. Due to the high computational costs when considering many different parameter configurations, hyperparameter tuning was performed by averaging the performances over the first 10 validation iterations for each simulation setting and each parameter setting. The setting which yielded the highest validation AUC across the average over these 10 repetitions was chosen as the fixed setting for the particular simulation setting.

The standard approach for testing the association considers the GRS as a predictor in a conventional regression model [2]. For binary outcomes, the logistic regression model is fitted on the test data. The logistic regression model maps the linear predictor with the logistic function from $(-\infty, +\infty)$ to $(0, 1)$. Thus, the GRS (probability estimates) are transformed to the scale of the linear predictor by applying the inverse of the logistic function, the logit function. In summary, the univariate association model

$$\operatorname{logit}(\mathbb{P}(Y = 1 \mid \mathrm{GRS})) = \beta_0 + \beta_1 \cdot \mathrm{GRS} \tag{5}$$

is constructed using

$$\left\{ \left(\mathrm{GRS}(\boldsymbol{x}), y\right) := \left(\operatorname{logit}(\mathrm{GRS}_{\mathrm{raw}}(\boldsymbol{x})), y\right) \;\middle|\; (\boldsymbol{x}, y) \in \mathcal{D}_{\mathrm{test}} \right\}$$

for raw risk predictions of the fitted GRS model $\mathrm{GRS}_{\mathrm{raw}}$.

For statistically assessing this association, we conducted Wald tests testing the alternative that the GRS is associated with the response. Based on these test results, we estimated the statistical power and the type I error rate for analyzing and comparing the ability of properly recognizing signals in the genetic data by the GRS construction procedures. The statistical power, which is given by the probability that the GRS is correctly recognized as influential on the response, can be estimated by the fraction of logistic models with statistically significant predictors under all cases which rely on theoretically influential genetic data. The type I error rate, i.e., the false positive rate, can be estimated by the fraction of significantly recognized GRS under all cases in which the response and the predictors are actually independent.

To compare the predictive strength of GRS, which is probably most relevant, we calculated the area under the curve (AUC) with respect to the receiver operating characteristic (ROC). This metric offers two main advantages over classification measures such as the accuracy, sensitivity, or specificity. First, it does not depend on the classification threshold which perhaps should be tuned. Second, the AUC can handle imbalanced data sets due to simultaneously regarding sensitivity and specificity. Moreover, the AUC has an intuitive interpretation as the probability that a random observation from the entire population of cases yields a higher risk estimate than a randomly chosen control from the population [52].

Additionally, we evaluated the classical classification metrics accuracy, sensitivity, and specificity. In particular, we performed hard classifications on the resulting logistic regression model containing the GRS using a classification threshold of 0.5, i.e., classifying an observation as a case if it is predicted that the probability of being a case is higher than the probability of being a control and classifying an observation as a control otherwise. Using these classifications, the overall accuracy, sensitivity, and specificity as defined, e.g., in Alberg et al. [53] were evaluated. The accuracy was not explicitly adjusted for the prevalence, since we generated approximately case-control-balanced data sets in the simulation study, thus, yielding a prevalence of 50%. However, the main purpose of GRS does not lie in hard classifying observations as cases or controls. Instead, GRS are used for estimating individual risks, e.g., in precision medicine or for uncovering biological mechanisms involved in the development of diseases. Therefore, a metric such as the AUC which simultaneously considers different sensitivities and specificities seems to be preferable in the evaluation of the performance of GRS.

Hyperparameter optimization

Certain statistical learning procedures require the optimization of hyperparameters using independent validation data sets. This also holds true for the algorithms considered in this article. Table 4 lists the regarded hyperparameter configurations, where each possible combination of these parameters has to be considered in the parameter tuning. A description of each of these parameters is given in Additional file 1: Section 2. For random forests, we fixed the number of total trees grown to 2000, which is a sufficiently large number of trees in our applications, since in preliminary experiments, we could observe that the validation AUC converged using smaller amounts of trees. Analogously, we fixed the number of bagging iterations for logic bagging to 500. The cooling schedule in logic regression was configured manually by observing the cooling behavior for different settings and choosing a start temperature and end temperature such that around 90% of the proposed models were accepted at the beginning of the algorithm and close to no models were accepted when approaching the end temperature. The amount of simulated annealing iterations was fixed to 500000. The regularization parameter λ for the

Table 4 Regarded hyperparameter settings

Algorithm	Hyperparameter	Considered realizations
Random forests & random forests VIM	mtry	$\lfloor \begin{pmatrix} 0.5 & 1 & 2 \end{pmatrix} \cdot \lfloor \sqrt{p} \rfloor \rfloor$
	min.node.size	$\lfloor \begin{pmatrix} 0.01 & 0.05 & 0.1 \end{pmatrix} \cdot N \rfloor$
	num.trees	2000
Logic regression & logic bagging	ntrees	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$
	nleaves	$\begin{pmatrix} 1 & 2 & \ldots & 9 & 10 \end{pmatrix}$ (Simulation studies)
		$\begin{pmatrix} 1 & 2 & \ldots & 19 & 20 \end{pmatrix}$ (Real data application)
Logic regression	Cooling schedule	Experimental
	Simulated annealing iterations	500000
Logic bagging	Bagging iterations	500
Elastic net	α	$\begin{pmatrix} 0.5 & 0.75 & 0.9 & 0.99 \end{pmatrix}$
	λ	Cross-validation

The mentioned hyperparameter names are the names of the corresponding arguments in the respective software packages. For a description of the parameters, see Additional file 1: Section 2

elastic net was automatically chosen by employing cross-validation in the respective fitting processes and selecting the value which minimizes the loss.

For each considered statistical learning method, a more detailed workflow for tuning and training the respective models is depicted in Additional file 1: Section 3.

Results of the simulation studies

Marginal genetic effects

Figure 3 summarizes the AUC for each of the 27 regarded settings in the main effects simulation scenario. In Additional file 1: Fig. S2, corresponding asymptotic 95% confidence intervals are depicted. Most notably, logic bagging leads in almost every scenario to the highest AUC. For strong effects and large data sets, ordinary logic regression induces similar or even better results which are comparable to the true underlying model. Especially for weak effects, ordinary random forests yields comparably high values for the AUC. Unsurprisingly, random forests with a prior variable selection is more effective in relation to the other procedures when considering a higher amount of statistical noise. For less noisy data, random forests VIM cannot compete with the other tree-based methods and shows high variations. The elastic net yields inferior results for large data sets and large effect sizes and also has difficulties detecting a signal for the more challenging scenarios, i.e., for small odds ratios and low observation counts.

The analyses of power resemble the results of the AUC comparison and are depicted in Additional file 1: Fig. S3. The type I error rates for the tree-based methods seem to randomly scatter around the prespecified significance level of 5%. However, the elastic net induces type I error rates of around two percent and is, therefore, quite conservative. The corresponding type I error rates are shown in Additional file 1: Fig. S4.

In Additional file 1: Figs. S5–S7, the results for the accuracy, sensitivity, and specificity are depicted. The accuracies resemble the results of the AUC evaluation, while the sensitivities and specificities do not show a clear pattern between the evaluated methods. These figures also show that, for increasing odds ratios, the specificities increase while the sensitivities decrease.

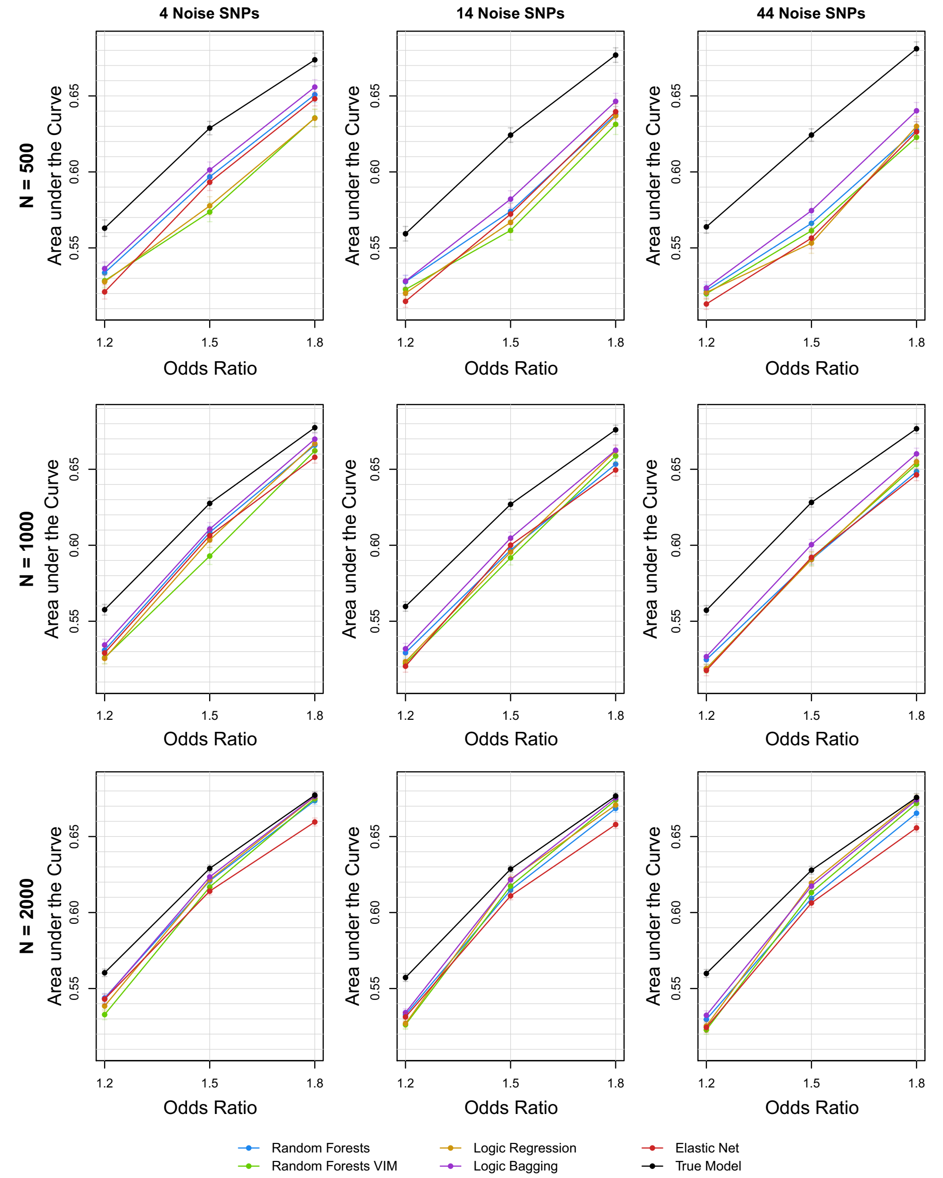

Fig. 3 Mean AUC for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the first simulation scenario considering marginal effective SNPs evaluated on the test data

We also evaluated the GRS on the training data itself to compare the degrees of overfitting. Here, ordinary random forests leads to the severest overfitting. For data with high statistical noise and small effect sizes, its AUC almost reaches 100% compared to the true AUC of around 56%. The other tree-based algorithms also induce higher training AUCs than the true model, but not larger than random forests. In particular, a prior variable selection can indeed reduce the intensity of overfitting. The elastic net yields in most cases the lowest values for the AUC closely following the AUCs of the true model. Taking the test data analyses into account, this indicates

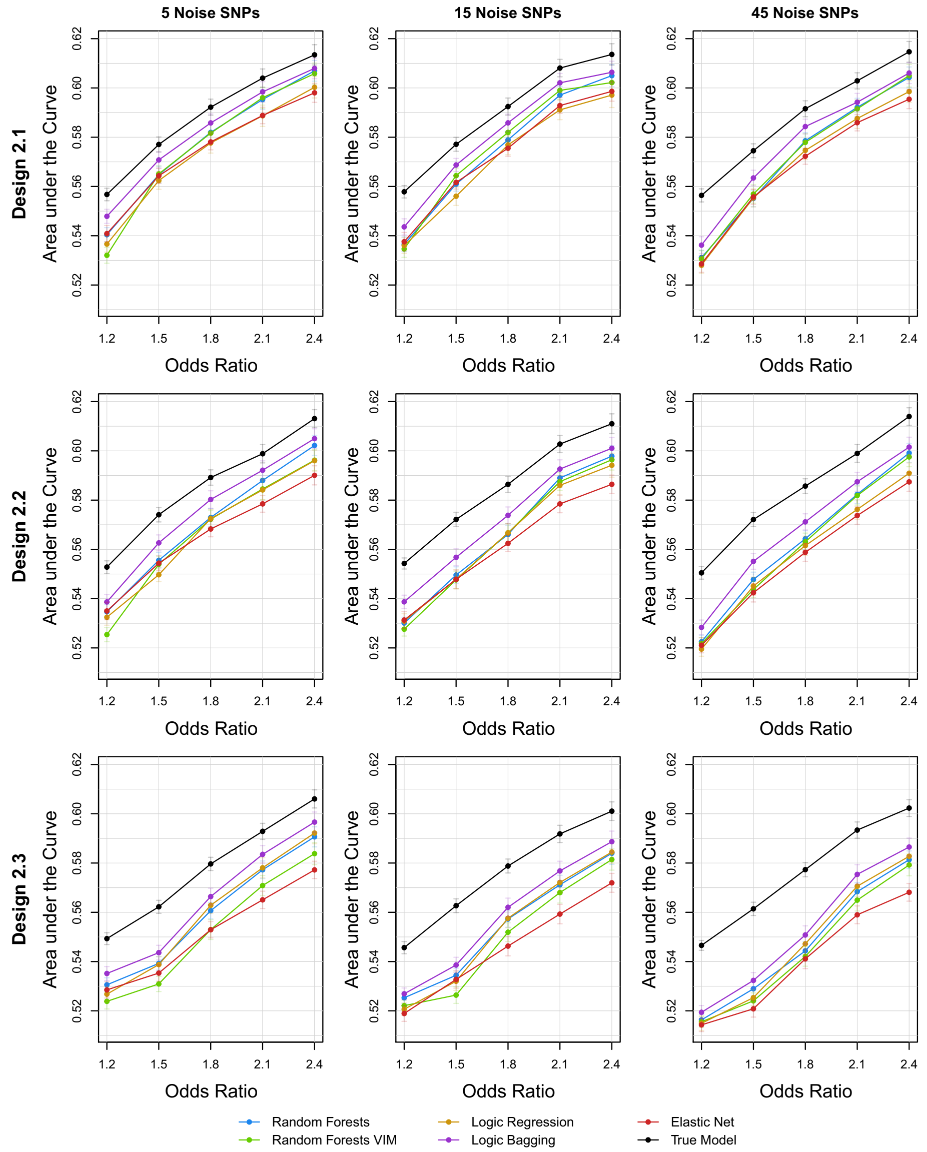

Fig. 4 Mean AUC for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the second simulation scenario incorporating interactions of SNPs evaluated on the test data. The Designs 2.1, 2.2, and 2.3 describe the scenarios where both interacting SNPs also exhibit marginal effects, only one of both SNPs shows a marginal signal or none of them induce a main effect, i.e., (j, k) = (1, 2), (1, 4), or (4, 5) in Eq. (3), respectively

a mixture of underfitting and slight overfitting of the elastic net. The training data results can be found in Additional file 1: Fig. S8.

Dominant interaction effects of SNPs

For the analysis of the scenarios with influential interaction terms, the performances of the statistical learning procedures measured by the AUC are shown in Fig. 4. Additionally, asymptotic 95% confidence intervals can be found in Additional file 1: Fig. S9. Similar to the main effects scenarios, logic bagging induces in each scenario the highest values of the AUC. Also as in the other settings, random forests VIM does not gravely suffer from noisy data compared to standard random forests, but cannot severely outperform its ordinary counterpart. Random forests itself seems to be the second-best performing method with an almost steady but close distance to logic bagging. Interactions of variables without marginal effects seem to be less of an issue to conventional logic regression, since for Design 2.3 and larger interaction effect sizes, logic regression achieves comparable AUCs to random forests. For weak interaction effects, the elastic net can yield comparative results to random forests and the logic regression. Nonetheless, increasing the interaction effect also increases the discrepancy between the tree-based approaches and the elastic net.

The results of the corresponding power and type I error analyses can be found in Additional file 1: Figs. S10 and S11. As in the previous simulation scenario, the comparison of the estimates of the statistical power resembles the corresponding analyses of the AUC. Again, the type I error rates for the tree-based methods seem to randomly scatter around 5%, whereas the elastic net leads to substantially lower error rates.

The results for the accuracy, sensitivity, and specificity can be found in Additional file 1: Figs. S12–S14. Similar to the marginal effects simulation scenario, the comparisons of the mean accuracy resemble the results of the AUC evaluation. The other two metrics sensitivity and specificity do not yield clear patterns between the considered procedures.

Evaluations of the GRS on the training data reveal again that conventional random forests seems to induce the severest overfitting. The results of these training data set applications are summarized in Additional file 1: Fig. S15.

Gene-environment interactions

Figure 5 depicts the predictive performances of the statistical learning procedures for the 20 settings in the GxE interaction simulation scenario. Corresponding asymptotic 95% confidence intervals are shown in Additional file 1: Fig. S16. In contrast to the previous scenario, a true unique GRS model does not exist, since the GRS is based only on the genetic data while the true model of this scenario also consists of environmental covariables. Similar to the gene-gene interaction scenario, logic bagging leads in each setting to the highest AUCs. Throughout all settings in this simulation scenario, logic regression seems to be the second best performing method yielding AUCs closely below the AUCs of logic bagging. Random forests and random forests VIM induce very similar results such that there is no clear pattern between these two methods. For weak GxE interaction effects, the elastic net induces comparably poor results. However, for increasing GxE interaction effects, the discrepancy between random forests and elastic net decreases such that, for an odds ratio of 2.4, the elastic net yields slightly higher AUCs than random forests which are, however, still below the AUCs of logic bagging.

The correlation ρ of the two continuous variables does not seem to affect the GRS performance in this simulation scenario. Nonetheless, the overall performance in Design 3.1 is higher than the performance in Design 3.2. This phenomenon can be explained by the absence of a marginal effect of the GxE interacting SNP in Design 3.2 complicating the identification of this SNP.

For this simulation scenario, the statistical power for all considered methods and simulation settings was equal to 100%. Similar to the previous scenarios, the elastic net

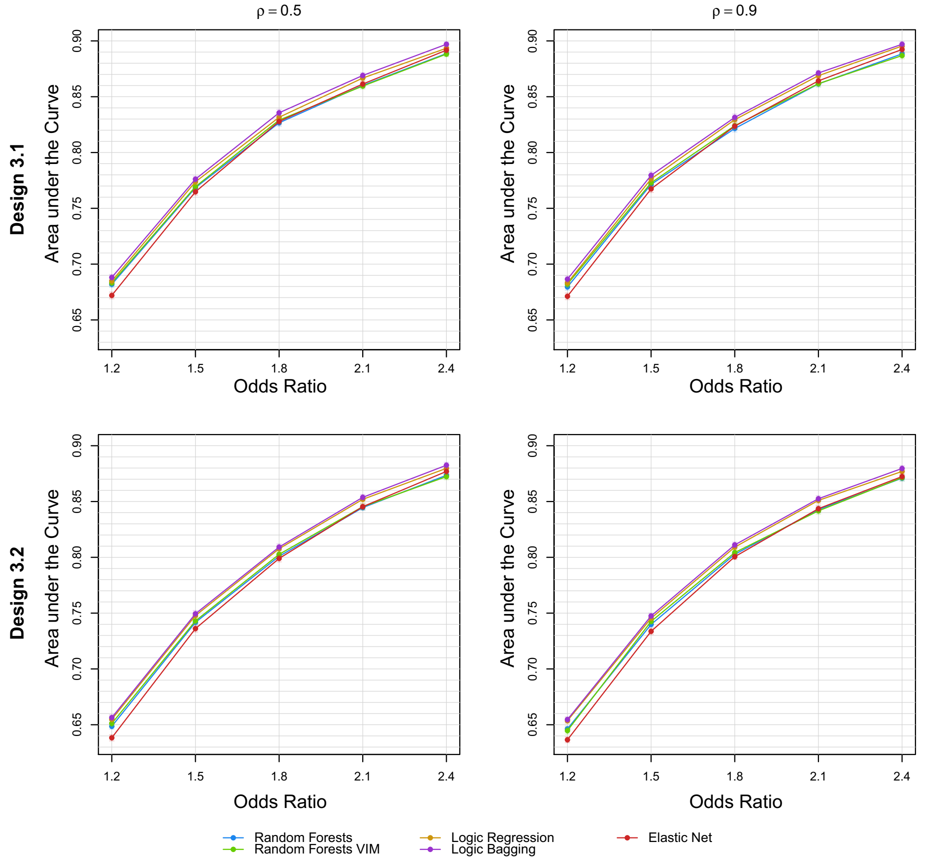

Fig. 5 Mean AUC for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the third simulation scenario incorporating continuous input variables evaluated on the test data. The Designs 3.1 and 3.2 describe the scenarios where the GxE interacting SNP also exhibits a moderate marginal effect or where it does not induce a main effect, i.e., j = 2 or 5 in Eq. (4), respectively

seems to be more conservative as it induces lower type I error rates than the tree-based methods. The estimated type I error rates can be found in Additional file 1: Table S1.

In Additional file 1: Fig. S17–S19, the results for the accuracy, sensitivity, and specificity are depicted. Similar to the power analyses, the mean accuracies of the considered methods are almost identical in each simulation setting. However, for weak GxE interaction effects, the elastic net seems to induce the lowest mean accuracies. The results for the other two metrics, the sensitivity and the specificity, are also very similar.

Training data evaluations reveal again that conventional random forests tends to induce the severest overfitting. The training data results are depicted in Additional file 1: Fig. S20.

Comparison considering binary SNP codings

Additionally to considering the standard way of specifying the input variables for the different methods, we also evaluated the GRS construction approaches using the binary $\{0, 1\}$ SNP coding for each method and not exclusively for logic regression and logic bagging. The detailed results for the $\{0, 1\}$ SNP coding and the respective simulation scenarios are depicted in Additional file 1: Figs. S21–S23.

In comparison to using the $\{0, 1, 2\}$ coding, the performance of random forests and random forests VIM decreases. This is not very surprising, since, as pointed out in the methodological description, decision trees and random forests consider the dominant and recessive modes of inheritance when using the $\{0, 1, 2\}$ coding. Thus, using the $\{0, 1\}$ coding doubles the number of input variables without supplying more information to random forests. The increase in the number of input variables complicates identifying the ideal splits when using typical settings for the hyperparameter *mtry*.

For the elastic net, the performance increases when employing the $\{0, 1\}$ coding instead of the conventional $\{0, 1, 2\}$ coding such that, in the marginal effects simulation scenario and in the GxE interaction scenario, the elastic net yields similar results as logic bagging when considering settings with stronger genetic effects. Nonetheless, in the gene-gene interaction simulation scenario for the Designs 2.2 and 2.3 in which at least one interacting SNP does not exhibit a marginal effect, the elastic net with the $\{0, 1\}$ SNP coding still induces inferior AUCs compared to logic bagging.

Real data application

We also compared the GRS construction approaches using a real data set from a German cohort study, the SALIA study (**St**udy on the Influence of **A**ir Pollution on **L**ung, **I**nflammation and **A**ging) [54], which included in total 4874 women that were at their first examination between 54 and 55 years old. The participants were recruited in 1985-1994 from highly industrialized areas and less industrialized areas in North-Rhine Westphalia, Germany. In 2006, a follow-up questionnaire was completed by 4027 women which contained questions about the diagnosis of certain diseases. In a further follow-up clinical examination conducted in 2007-2010, genetic data was also gathered. Here, we considered a data set consisting of 517 women from the SALIA study for which the presence of rheumatic diseases and genetic data are available. Furthermore, information about the exposure to specific air pollutants, i.e., nitrogen dioxide (NO_2), nitrogen oxide [nitrogen monoxide NO and nitrogen dioxide NO_2] (NO_x), particulate matter with an aerodynamic diameter of $\leq 2.5\mu m$ or $\leq 10\mu m$ ($PM_{2.5}$ or PM_{10}), particulate matter with diameters of $2.5 - 10\mu m$ (PM_{coarse}), and the reflectance of $PM_{2.5}$ filters ($PM_{2.5\ absorbance}$), is available at the time of performing the examinations in 2008. The assessment of the exposure to air pollution was conducted as part of the ESCAPE (**E**uropean **S**tudy of **C**ohorts for **A**ir **P**ollution **E**ffects) project using land-use regression models [55, 56]. We used these air pollution exposures to assess GxE interactions. Information on covariables such as the BMI (body mass index), age, education status, smoking status, or workplace exposure for adjusting the final models is also available. In the questionnaire, it was asked whether any rheumatic disease was diagnosed. Thus, we considered prevalent rheumatic diseases as outcome in our analyses. Details on the SALIA study and the assessment of air pollution in this study are given by Krämer et al. [57] and Hüls et al. [58].

Selection of relevant genetic factors

In order to construct proper GRS for genes potentially having an impact on the development of rheumatic diseases, we selected several genes which showed to be influential in a literature research. For the selection of relevant genes, we mainly focused on rheumatoid arthritis, since it is the most common rheumatic disease besides osteoarthritis [59–61].

In around 70% to 90% of rheumatoid arthritis patients, anti-citrullinated peptide antibodies (ACPA) can be detected [62]. For ACPA-positive rheumatoid arthritis, many identified genetic associations belong to the human leukocyte antigen (HLA) class II complex [63]. Thus, we selected genes from the HLA class II complex for which associations with rheumatoid arthritis have been detected. In particular, we chose the HLA-DRB1 gene which presumably explains a large portion of the heritability of rheumatoid arthritis in the HLA class II complex [63–66]. Furthermore, we included the HLA-DPB1 and HLA-DOA genes which also might influence the risk of developing rheumatoid arthritis [66–68].

Since we started by including all available SNPs within the respective genes, 385 SNPs from the three genes formed our basis which we reduced by exploiting high states of LD. Using PLINK version 1.9 [69, 70], we performed LD-based clumping [71] (considering $r^2 = 0.5$). This procedure resulted in 72 tag SNPs which were used to construct the GRS.

We also constructed genome-wide GRS based on a recent meta-analysis of GWAS regarding rheumatoid arthritis [72]. In this meta-analysis, only non-HLA loci were considered in contrast to the gene-based selection. 70 of the proposed SNPs were available in our data and were used to fit the GRS models.

Gene-environment interaction analysis

Additionally, we also analyzed GxE interaction effects. For the risk of developing ACPA-positive rheumatoid arthritis, GxE interactions between HLA class II alleles and smoking have been discovered [73, 74]. It might be of interest if traffic-related air pollution also interacts with genetic risk factors in the development of rheumatoid arthritis. Thus, our logistic regression models for the evaluation of GRS have the shape

$$\operatorname{logit}(\mathbb{P}(Y=1)) = \beta_0 + \beta_1 \cdot \mathrm{GRS} + \beta_2 \cdot E + \beta_3 \cdot \mathrm{GRS} \cdot E + \sum_{i=1}^{l} \gamma_i \cdot C_i \tag{6}$$

for the environmental variable E and covariables $C_1, \ldots, C_l$.

The selection of potential relevant covariables was performed in two steps. First, we applied a stepwise logistic regression with the AIC (Akaike information criterion) as the selection measure. This lead to the inclusion of the age, the BMI, the current smoking status, and the former smoking status. Next, we regarded this selection of variables in the final models jointly with the GRS and air pollutants. We excluded covariables which worsened the models, i.e., which lead to lower AUCs. After this procedure, only the age was left.

Analysis of association and predictive strength

The analysis was conducted in a repeated train-test split scheme. For 100 repetitions, we randomly divided the whole data set into 50% training data and 50% test data similar to Hüls et al. [11]. The respective training data sets were further randomly divided into 75% training data for hyperparameter tuning and 25% validation data (for the considered values of the hyperparameters, see "Section Hyperparameter optimization"). The best

Table 5 Descriptive statistics of the regarded data set from the SALIA study stratified according to the status of rheumatic diseases

Variable		Controls	Cases
N		394	123
Mean age	[years] ± sd	70.87 ± 3.16	71.50 ± 2.96
Mean BMI	[kg/m^2] ± sd	26.42 ± 3.93	27.46 ± 3.86
N Currently smoking		21 (5.44%)	5 (4.07%)
N Formerly smoking		61 (15.80%)	15 (12.20%)
Mean pack-years of smoking	[years] ± sd	3.78 ± 10.92	2.85 ± 9.25
Mean NO$_2$	[µg/m^3] ± sd	26.66 ± 7.34	27.94 ± 7.69
Mean NO$_x$	[µg/m^3] ± sd	41.34 ± 17.71	44.10 ± 17.68
Mean PM$_{10}$	[µg/m^3] ± sd	26.99 ± 2.16	27.39 ± 2.42
Mean PM$_{\text{coarse}}$	[µg/m^3] ± sd	9.52 ± 1.66	9.81 ± 1.84
Mean PM$_{2.5}$	[µg/m^3] ± sd	17.94 ± 1.38	18.23 ± 1.50
Mean PM$_{2.5\ \text{absorbance}}$	[µg/m^3] ± sd	1.47 ± 0.46	1.58 ± 0.59

Table 6 Median p-values of the Wald tests for univariate models only including the GRS built on the SALIA data set

Algorithm	Median *p* value
Random forests	0.018
Random forests VIM	0.167
Logic regression	0.353
Logic bagging	0.021
Elastic net	0.512

performing hyperparameter setting across the average of these 100 validation iterations was chosen.

Results of the real data application

A descriptive summary of the most important variables gathered in the data set from the SALIA study is given by Table 5. Most noticeably, we considered an unbalanced data set with 394 controls and 123 cases considering prevalent rheumatic diseases.

Univariate regression models

In the analysis of the data of the SALIA study, Table 6 summarizes the median p-values of GRS analyzed in univariate regression models as in Eq. (5). When testing the influence of the GRS on the risk of developing rheumatoid arthritis, conventional random forests and logic bagging are the only models achieving significance at a significance level of 5% for at least 50% of the evaluations.

Figure 6 summarizes the test AUC values for the tree-based statistical learning procedures and elastic net induced by univariate regression models only based on the GRS. For the gene-based approach, most noticeably, random forests and logic bagging yield the highest AUCs where random forests achieves a slightly better performance than logic bagging. Ordinary logic regression and random forests with a prior variable selection induce similar results which cannot compete with conventional random forests and

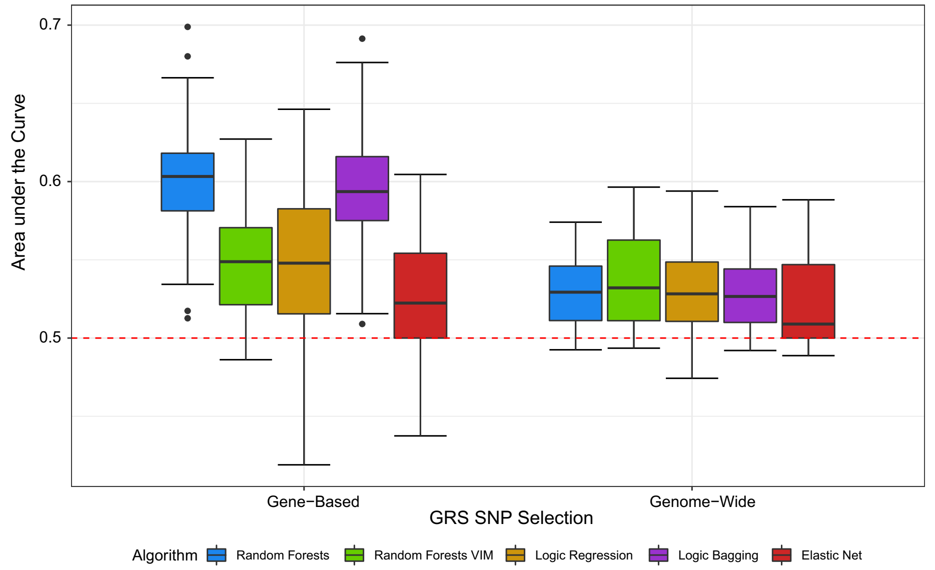

Fig. 6 AUC for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the application to data from the SALIA study evaluated on the test data. Results for single unadjusted models also considering the alternative genome-wide construction approach

logic bagging. However, the elastic net yields the lowest AUCs. Here, the lower quartile of the AUCs yielded by the elastic net reaches 50%, i.e., the predictive performance of a (non-informative) constant classifier.

In addition to gene-based GRS, we also constructed genome-wide GRS based on a recent GWAS meta-analysis regarding rheumatoid arthritis [72]. A specific comparison of the predictive power between the gene-based and genome-wide approaches is summarized in Fig. 6. However, for the genome-wide selection of SNPs, barely a signal can be observed in our sample as the AUCs on the test data sets were close to 50%. Thus, the genome-wide GRS construction approach was not included in subsequent analyses. The inferior predictive performance compared to the gene-based selection is possibly caused by the exclusion of HLA genes in the underlying meta-analysis. Nonetheless, the elastic net induces the lowest values for the AUC compared to the tree-based methods which is in line with our previous experiments. In contrast to the gene-based approach, random forests VIM yields a predictive power that can compete with ordinary random forests and logic bagging.

Gene-environment interaction analysis

In the final adjusted models of the form as in Eq. (6), we regarded each air pollutant indicator separately and included the respective GxE interaction term. Neither the GRS themselves nor the GxE interaction terms are significant at a significance level of 5%. The concrete median p-values of the 100 repetitions for the final adjusted models can be found in Additional file 1: Table S2.

Figure 7 depicts the predictive performance of the considered statistical learning algorithms for the induction of gene-based GRS in multivariate regression models. Analogously to the univariate analysis, random forests and logic bagging yield the highest

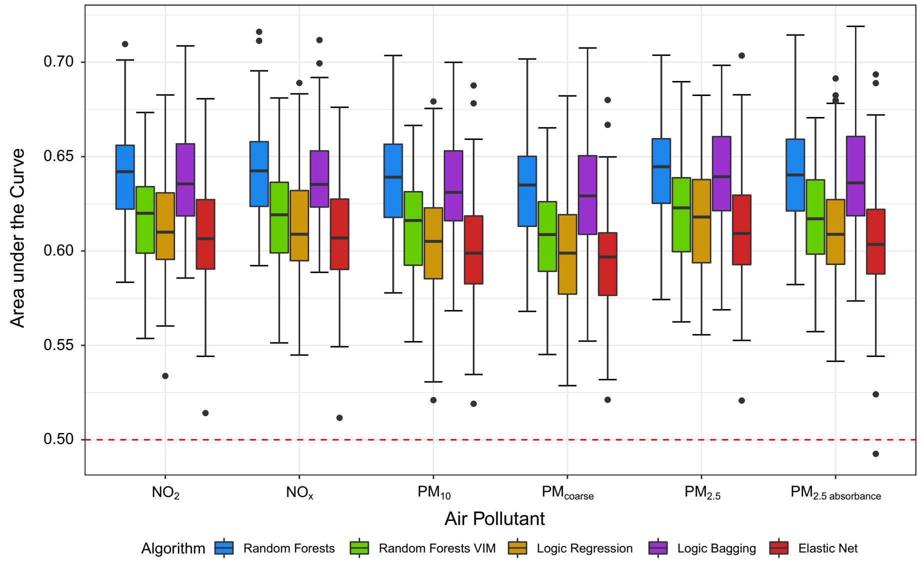

Fig. 7 AUC for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the application to data from the SALIA study evaluated on the test data. Results for the final age-adjusted models with different air pollution indicators

predictive power where the overall best values are reached for $PM_{2.5}$. For this air pollutant, random forests achieves the best performance. The elastic net, random forests VIM, and logic regression yield similar performances which, again, cannot compete with random forests and logic bagging.

We also evaluated the GRS on the training data sets themselves. The best performing procedures random forests and logic bagging tend to heavily overfit the data as can be seen by the high discrepancy between the test and the training data analyses. These two algorithms achieve training AUCs of nearly 100% whereas the other methods lead to more homogeneous results. The corresponding AUCs can be found in Additional file 1: Fig. S24.

Smoking is a major risk factor for rheumatoid arthritis [75]. As can be seen in Table 5, the fractions of current smokers and former smokers in the excerpt from the SALIA study are higher among controls than among cases which is in contradiction to the literature. Since only 19.7% of the study participants in the data excerpt are current or former smokers, we conducted a sensitivity analysis excluding all current and former smokers from the data. Again, we are not able to identify any significant GxE interactions. The resulting AUCs are very similar to the former analysis. Random forests and logic bagging yield the highest test AUC values, whereas elastic net induces substantially lower values. The concrete results can be found in Additional file 1: Fig. S25.

Discussion

In this analysis, we evaluated tree-based statistical learning approaches for the construction of GRS. We used the elastic net as a reference model and analyzed the tree-based statistical learning methods in a simulation study considering several scenarios, focusing on marginal and epistatic genetic effects, respectively. To confirm our findings, we constructed and assessed GRS on a real data set from the German SALIA cohort study.

As our analyses showed, a modification of logic regression, namely logic bagging, was able to outperform the reference GRS construction procedure, the elastic net, in almost every scenario of the simulation study.

Similarly, logic bagging lead to a comparably strong predictive performance in the real data application. Logic regression could only compete when considering large effect sizes in the simulation studies and yielded inferior results in the analysis of the SALIA data. This indicates that logic regression fits highly variant models which can indeed benefit from a variance reduction via an ensemble approach like bagging. For larger genetic effects, bagging does not seem to be necessary due to a more consequent identification of the underlying signal.

Random forests lead to the best predictive performance on the real data set. Considering the simulation study, in a likewise comparable scenario, i.e., small data sets, low marginal genetic effects, and higher amounts of statistical noise, random forests could induce comparably high values for the AUC as well. In the analysis of marginal genetic effects, random forests' performance decreased for increasing amounts of noise. This phenomenon can be partly explained by the random selection scheme of predictors for partitioning. The input variables are drawn with equal probabilities without replacement. Therefore, considering the setting with 44 noise SNPs in the first simulation scenario, in a decision tree branch where already three of the six influential SNPs and no noise are included, the probability of regarding one of the three remaining influential SNPs for the next split with the standard setting $mtry = \lfloor\sqrt{50}\rfloor = 7$ is about only 39%. Thus, choosing a set of SNPs containing only statistical noise is more likely in this case. We also allowed higher settings for *mtry* in the hyperparameter optimization as could be seen in Table 4. For higher amounts of statistical noise, the higher setting for *mtry* could in fact increase the performance of random forests.

A related issue was the high amount of overfitting by random forests which could be observed in all three simulation scenarios as well as in the real data application. We addressed this by considering minimum terminal node sizes of up to 10% of the number of observations in each leaf and by performing a prior variable selection based on variable importance measures. The former solution, i.e., the tuning of the minimum node size, was important to optimize the performance on the general population, since the standard setting is set to one observation for classification trees. However, for appropriate probability estimates, Malley et al. [35] recommend choosing 10% of the total sample size.

The latter approach, i.e., the usage of random forests VIM, needed higher amounts of statistical noise and stronger marginal genetic effects to achieve test data performances comparable to random forests. Nonetheless, this alternative approach could substantially reduce the amount of overfitting in any case. Presumably caused by weak individual genetic effects, random forests VIM yielded an inferior predictive performance compared to ordinary random forests on the application to the SALIA data. However, in the analyses conducted by Speiser et al. [76], the random forests VIM approach utilizing the Boruta variable selection was able to yield lower error rates than conventional random forests. Thus, studies specifically comparing random forests variable selection procedures with conventional random forests in low signal-to-noise ratio scenarios, such as applications considering SNP data, might be beneficial.

The reference procedure, the elastic net, could not compete with logic bagging and random forests when considering stronger gene-gene interaction effects. Even for solely marginal genetic effects, the regularization procedure had difficulties achieving AUCs as high as the ones of logic bagging. However, for strong GxE interaction effects, the elastic net could induce similar predictive performances as random forests. Before deciding to choose the penalty parameter λ based on the minimum cross-validation error, we evaluated the elastic net based on the maximum λ which yielded a cross-validation error in the range of one standard error of the minimum error. This approach is also recommended by Waldmann et al. [77] for GWAS-level amounts of SNPs and used by Hüls et al. [49] for the construction of GRS. However, in our applications including both the simulation study and the real data application, the elastic net had difficulties recognizing a signal at all with this approach which was presumably caused by high errors in general. Thus, we chose the minimizing λ which enhanced our fitted elastic net models.

In practice, the conventional $\{0, 1, 2\}$ SNP coding is utilized when constructing GRS with regularized regression approaches such as the elastic net [11, 16]. Thus, we focused on this standard procedure in our analyses, which lead to comparatively weak performances. However, when splitting each considered SNP into two binary variables, i.e., when using the binary $\{0, 1\}$ SNP coding also for the elastic net, its performance in the simulation study increased due to now being able to differentiate between the dominant and recessive modes of inheritance. Therefore, the results for the $\{0, 1\}$ SNP coding suggest that it might be preferable to employ the $\{0, 1\}$ coding when fitting GRS using the elastic net. Nonetheless, logic bagging still yielded higher predictive performances than the elastic net in the gene-gene interaction simulation scenario when considering the $\{0, 1\}$ coding for all procedures.

The most important advantage of the tree-based methods regarded in this article is to not being restricted to model assumptions such as linearity, i.e., being able to autonomously detect gene-gene interactions. The assumption of oversimplified genetic architectures in linear models might be the main cause for random forests and logic bagging outperforming the elastic net in most analyses. However, it is well known that gene-gene interactions also play a role in the heritability of diseases [8, 9].

Another practically interesting question would be, how well the introduced tree-based methods can construct GRS for significantly larger amounts of SNPs, e.g., when using a broader SNP selection from GWAS. Winham et al. [22] found in their studies that for increasing amounts of SNPs, the identification of interactions becomes more difficult for random forests. For logic regression, with increasing amounts of explanatory variables, the amount of possible states increases linearly, therefore, requiring more simulated annealing iterations and generally deeper greedy searches and, hence, increasing the model fitting time. This model building time must be further increased when considering higher values for the parameters of maximum trees and maximum leaves which is reasonable due to potentially more influential predictors for more total input variables.

Unsurprisingly, elastic net models could be fitted and evaluated in the least amount of time due to their simplicity compared to the considered tree-based models. Random forests with 2000 trees could be fitted and evaluated in less than 10 s in most cases. Random forests VIM needed slightly more time which was to be expected. Logic bagging models needed more time, however, conventional logic regression models utilizing simulated annealing as search procedure consumed the most amount of time and needed up to 1 minute for fitting and evaluating the GRS. In Additional file 1: Fig. S1, the concrete times for the third simulation scenario are depicted.

For increasing odds ratios, the measured sensitivity decreases in the marginal effects and gene-gene interaction effect simulation scenarios, which does not seem to be plausible at first glance. However, this phenomenon can be explained by the data structure considered in this analysis and the requirement to dichotomize the risk predictions into two classes for estimating the sensitivity and specificity. For constructing GRS, discrete input variables, more exactly SNPs exhibiting three different outcomes, are used. Thus, the constructed and possibly true underlying GRS also follow a discrete pattern depending on the SNP setting. For the marginal effects simulation scenario, there are 7 distinct GRS values in the true underlying model due to Eq. (2). In Additional file 1: Fig. S26, a corresponding GRS distribution is depicted. Due to the additivity in this model, the GRS just below 0.5 occurs in approximately 30% of all observations. Therefore, dichotomizing the GRS at 0.5 leads to classifying only 35% of all observations as cases which explains the low sensitivity in this setting. Lowering the classification threshold to a value such as 0.45 shifts the issue to the specificity, since, in this case, only 35% of all observations will be classified as controls. Thus, the sensitivities and specificities determined in this analysis need to be interpreted with caution because of the discrete nature of the considered input variables.

In our real data application, we analyzed a relatively small data set containing 517 observations with only 123 cases. The missing balance as well as the comparably low sample size complicated meaningful analyses, especially when considering the need for splitting the data set into training and test data sets. Generally, important covariates such as the smoking status and the BMI were not included in the final models due to lowering the predictive performance. This decrease in performance was presumably caused by the low sample size and amount of cases yielding unintuitive statistics such as the higher fraction of smokers among controls.

Conclusion

As our analyses on simulated as well as on real data showed, the tree-based statistical learning methods random forests and logic bagging can be valuable tools for constructing GRS. Especially when little prior knowledge about the gene-response relationships is available or if no appropriate external weights for the regarded disease or population are available, these two algorithms should also be taken into consideration when building GRS. Regardless of the presence of gene-gene interactions in the heritability of a certain disease, the discussed methods have the potential to outperform regularized linear methods.

Abbreviations

ACPA: Anti-citrullinated peptide antibody; AIC: Akaike information criterion; AUC: Area under the curve; BMI: Body mass index; CART: Classification and regression tree; GLM: Generalized linear model; GRS: Genetic risk score(s); GWAIS: Genome-wide association interaction study; GWAS: Genome-wide association study; GxE: Gene-environment; HLA: Human leukocyte antigen; IQR: Interquartile range; LD: Linkage disequilibrium; MAF: Minor allele frequency; ROC: Receiver operating characteristic; SNP: Single nucleotide polymorphism; VIM: Variable importance measure.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-022-04634-w.

Additional file 1. Further evaluation results, hyperparameter descriptions, and method workflows. Additional methodological descriptions and results for the simulation study and the real data application.

Additional file 1. Simulation study data generating code. R code for generating and accessing all data sets used in the simulation study.

Acknowledgements
Not applicable.

Authors' contributions
ML, HS, and CW designed the simulation study. ML, SK, CW, and TS conceived the analyses of the real data application. The simulation study and real data evaluations were conducted by ML. ML was the major contributor in writing the manuscript. All authors read and approved the final manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL. This work has been supported (in part) by the Research Training Group "Biostatistical Methods for High-Dimensional Data in Toxicology" (RTG 2624, Project R3) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation—Project Number 427806116). The SALIA follow-up study 2007–2010 was funded by the German Statutory Accident Insurance (DGUV) Grant No: 617.0-FP266 and the European Community's Seventh Framework Programme (FP 7/2007-2011) under the Grant Agreement Number 211250.

Availability of data and materials
All code for generating and accessing data for the simulation study is included in this published article as a supplementary information file (Additional file 2).

Declarations

Ethics approval and consent to participate
The study was conducted in accordance to the declaration of Helsinki. The SALIA cohort study has been approved by the Ethics Committees of the Ruhr-University Bochum and the Heinrich Heine University Düsseldorf. We received written informed consent from all participants.

Consent for publication
Not applicable.

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

Author details
[1]Mathematical Institute, Heinrich Heine University, Düsseldorf, Germany. [2]IUF – Leibniz Research Institute for Environmental Medicine, Düsseldorf, Germany.

Received: 29 October 2021 Accepted: 14 March 2022
Published online: 21 March 2022

References

1. Billings LK, Florez JC. The genetics of type 2 diabetes: what have we learned from GWAS? Ann N Y Acad Sci. 2010;1212(1):59–77.
2. Choi SW, Mak TSH, O'Reilly PF. Tutorial: a guide to performing polygenic risk score analyses. Nat Protoc. 2020;15(9):2759–72.
3. Dudbridge F. Power and predictive accuracy of polygenic risk scores. PLoS Genet. 2013;9(3):1–17.
4. Torkamani A, Wineinger NE, Topol EJ. The personal and clinical utility of polygenic risk scores. Nat Rev Genet. 2018;19(9):581–90.
5. Wray NR, Lin T, Austin J, McGrath JJ, Hickie IB, Murray GK, et al. From basic science to clinical application of polygenic risk scores: a primer. JAMA Psychiat. 2021;78(1):101–9.
6. Thomas M, Sakoda LC, Hoffmeister M, Rosenthal EA, Lee JK, van Duijnhoven FJB, et al. Genome-wide modeling of polygenic risk score in colorectal cancer risk. Am J Hum Genet. 2020;107(3):432–44.
7. Kooperberg C, LeBlanc M, Obenchain V. Risk prediction using genome-wide association studies. Genet Epidemiol. 2010;34(7):643–52.
8. Gilbert-Diamond D, Moore JH. Analysis of gene–gene interactions. Curr Protocols Human Genet. 2011;70(1):1.14.1–1.14.12.
9. Ritchie MD, Van Steen K. The search for gene-gene interactions in genome-wide association studies: challenges in abundance of methods, practical considerations, and biological interpretation. Ann Transl Med. 2018;6(8):157.

10. Che R, Motsinger-Reif A. Evaluation of genetic risk score models in the presence of interaction and linkage disequilibrium. Front Genet. 2013;4:138.
11. Hüls A, Ickstadt K, Schikowski T, Krämer U. Detection of gene-environment interactions in the presence of linkage disequilibrium and noise by using genetic risk scores with internal weights from elastic net regression. BMC Genet. 2017;18(1):55.
12. Ottman R. Gene-environment interaction: definitions and study design. Prev Med. 1996;25(6):764–70.
13. Tibshirani R. Regression shrinkage and selection via the Lasso. J R Stat Soc Ser B (Methodol). 1996;58(1):267–88.
14. Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc Ser B (Stat Methodol). 2005;67(2):301–20.
15. Mavaddat N, Michailidou K, Dennis J, Lush M, Fachal L, Lee A, et al. Polygenic risk scores for prediction of breast cancer and breast cancer subtypes. Am J Human Genet. 2019;104(1):21–34.
16. Privé F, Aschard H, Blum MGB. Efficient implementation of penalized regression for genetic risk prediction. Genetics. 2019;212(1):65–74.
17. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.
18. Fu H, Zhang Q, Qiu G. Random forest for image annotation. In: Computer Vision—ECCV 2012. Berlin: Springer; 2012. p. 86–99.
19. Elagamy MN, Stanier C, Sharp B. Stock market random forest-text mining system mining critical indicators of stock market movements. In: 2018 2nd international conference on natural language and speech processing (ICNLSP); 2018. p. 1–8.
20. Hao M, Jiang D, Ding F, Fu J, Chen S. Simulating spatio-temporal patterns of terrorism incidents on the Indochina Peninsula with GIS and the random forest method. ISPRS Int J Geo-Inf. 2019;8(3):133.
21. Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees. Boca Raton: CRC Press; 1984.
22. Winham SJ, Colby CL, Freimuth RR, Wang X, de Andrade M, Huebner M, et al. SNP interaction detection with Random Forests in high-dimensional genetic data. BMC Bioinform. 2012;13(1):164.
23. Ruczinski I, Kooperberg C, LeBlanc M. Logic Regression. J Comput Graph Stat. 2003;12(3):475–511.
24. Schwender H, Ickstadt K. Identification of SNP interactions using logic regression. Biostatistics. 2007;9(1):187–98.
25. Kooperberg C, Ruczinski I. Identifying interacting SNPs using Monte Carlo logic regression. Genet Epidemiol. 2005;28(2):157–70.
26. Dinu I, Mahasirimongkol S, Liu Q, Yanai H, Sharaf Eldin N, Kreiter E, et al. SNP-SNP interactions discovered by logic regression explain Crohn's disease genetics. PLoS ONE. 2012;7(10):1–6.
27. Kruppa J, Ziegler A, König IR. Risk estimation and risk prediction using machine-learning methods. Hum Genet. 2012;131(10):1639–54.
28. Botta V, Louppe G, Geurts P, Wehenkel L. Exploiting SNP correlations within random forest for genome-wide association studies. PLoS ONE. 2014;9(4):1–11.
29. Gola D, Erdmann J, Müller-Myhsok B, Schunkert H, König IR. Polygenic risk scores outperform machine learning methods in predicting coronary artery disease status. Genet Epidemiol. 2020;44(2):125–38.
30. Badré A, Zhang L, Muchero W, Reynolds JC, Pan C. Deep neural network improves the estimation of polygenic risk scores for breast cancer. J Hum Genet. 2021;66(4):359–69.
31. Yoo W, Ference BA, Cote ML, Schwartz A. A comparison of logistic regression, logic regression, classification tree, and random forests to identify effective gene-gene and gene-environmental interactions. Int J Appl Sci Technol. 2012;2(7):268.
32. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. New York: Springer; 2009.
33. Li RH, Belford GG. Instability of decision tree classification algorithms. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining. New York: Association for Computing Machinery; 2002. p. 570–575.
34. Breiman L. Bagging predictors. Mach Learn. 1996;24(2):123–40.
35. Malley JD, Kruppa J, Dasgupta A, Malley KG, Ziegler A. Probability machines: consistent probability estimation using nonparametric learning machines. Methods Inf Med. 2012;51(1):74–81.
36. Provost F, Domingos P. Tree induction for probability-based ranking. Mach Learn. 2003;52(3):199–215.
37. Kursa MB, Rudnicki WR. Feature selection with the Boruta package. J Stat Softw. 2010;36(11):1–13.
38. Janitza S, Celik E, Boulesteix AL. A computationally fast variable importance test for random forests for high-dimensional data. Adv Data Anal Classif. 2018;12(4):885–915.
39. Altmann A, Toloşi L, Sander O, Lengauer T. Permutation importance: a corrected feature importance measure. Bioinformatics. 2010;26(10):1340–7.
40. Degenhardt F, Seifert S, Szymczak S. Evaluation of variable selection methods for random forests and omics data sets. Brief Bioinform. 2017;20(2):492–503.
41. 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):1–17.
42. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science. 1983;220(4598):671–80.
43. Kooperberg C, Ruczinski I. LogicReg: Logic Regression; 2021. R package version 1.6.3.
44. Schwender H, Tietz T. logicFS: Identification of SNP Interactions; 2020. R package version 2.10.0.
45. Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. Technometrics. 1970;12(1):55–67.
46. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J Stat Softw. 2010;33(1):1–22.
47. R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria; 2020. Available from: https://www.R-project.org/.
48. Schwender H, Fritsch A. scrime: Analysis of High-Dimensional Categorical Data Such as SNP Data; 2018. R package version 1.3.5.

49. Hüls A, Krämer U, Carlsten C, Schikowski T, Ickstadt K, Schwender H. Comparison of weighting approaches for genetic risk scores in gene-environment interaction studies. BMC Genet. 2017;18(1):115.
50. Li Q, Fallin MD, Louis TA, Lasseter VK, McGrath JA, Avramopoulos D, et al. Detection of SNP-SNP interactions in trios of parents with schizophrenic children. Genet Epidemiol. 2010;34(5):396–406.
51. Pan D, Li Q, Jiang N, Liu A, Yu K. Robust joint analysis allowing for model uncertainty in two-stage genetic association studies. BMC Bioinform. 2011;12(1):9.
52. Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology. 1982;143(1):29–36.
53. Alberg AJ, Park JW, Hager BW, Brock MV, Diener-West M. The use of "overall accuracy" to evaluate the validity of screening or diagnostic tests. J Gen Internal Med. 2004;19(5p1):460–465.
54. Schikowski T, Sugiri D, Ranft U, Gehring U, Heinrich J, Wichmann HE, et al. Long-term air pollution exposure and living close to busy roads are associated with COPD in women. Respir Res. 2005;6(1):152.
55. Beelen R, Raaschou-Nielsen O, Stafoggia M, Andersen ZJ, Weinmayr G, Hoffmann B, et al. Effects of long-term exposure to air pollution on natural-cause mortality: an analysis of 22 European cohorts within the multicentre ESCAPE project. Lancet. 2014;383(9919):785–95.
56. Eeftens M, Beelen R, de Hoogh K, Bellander T, Cesaroni G, Cirach M, et al. Development of land use regression models for PM2.5, PM2.5 absorbance, PM10 and PMcoarse in 20 European Study areas; results of the ESCAPE project. Environ Sci Technol. 2012;46(20):11195–205.
57. Krämer U, Herder C, Sugiri D, Strassburger K, Schikowski T, Ranft U, et al. Traffic-related air pollution and incident type 2 diabetes: results from the SALIA cohort study. Environ Health Perspect. 2010;118(9):1273–9.
58. Hüls A, Krämer U, Herder C, Fehsel K, Luckhaus C, Stolz S, et al. Genetic susceptibility for air pollution-induced airway inflammation in the SALIA study. Environ Res. 2017;152:43–50.
59. Vanhoof J, Declerck K, Geusens P. Prevalence of rheumatic diseases in a rheumatological outpatient practice. Ann Rheum Dis. 2002;61(5):453–5.
60. Jokar M, Jokar M. Prevalence of inflammatory rheumatic diseases in a rheumatologic outpatient clinic: analysis of 12626 cases. Rheumatol Res. 2018;3(1):21–7.
61. Sangha O. Epidemiology of rheumatic diseases. Rheumatology. 2000;39(suppl_2):3–12.
62. Song YW, Kang EH. Autoantibodies in rheumatoid arthritis: rheumatoid factors and anticitrullinated protein antibodies. QJM Int J Med. 2009;103(3):139–46.
63. Kampstra AS, Toes RE. HLA class II and rheumatoid arthritis: the bumpy road of revelation. Immunogenetics. 2017;69(8):597–603.
64. Clarke A, Vyse TJ. Genetics of rheumatic disease. Arthritis Res Therapy. 2009;11(5):1–9.
65. Eyre S, Bowes J, Diogo D, Lee A, Barton A, Martin P, et al. High-density genetic mapping identifies new susceptibility loci for rheumatoid arthritis. Nat Genet. 2012;44(12):1336–40.
66. Raychaudhuri S, Sandor C, Stahl EA, Freudenberg J, Lee HS, Jia X, et al. Five amino acids in three HLA proteins explain most of the association between MHC and seropositive rheumatoid arthritis. Nat Genet. 2012;44(3):291–6.
67. Jiang L, Jiang D, Han Y, Shi X, Ren C. Association of HLA-DPB1 polymorphisms with rheumatoid arthritis: a systemic review and meta-analysis. Int J Surg. 2018;52:98–104.
68. Okada Y, Suzuki A, Ikari K, Terao C, Kochi Y, Ohmura K, et al. Contribution of a non-classical HLA gene, HLA-DOA, to the risk of rheumatoid arthritis. Am J Human Genet. 2016;99(2):366–74.
69. Purcell S, Chang C. PLINK 1.9; 2021. Available from: www.cog-genomics.org/plink/1.9/.
70. Chang CC, Chow CC, Tellier LC, Vattikuti S, Purcell SM, Lee JJ. Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience. 2015;4:7.
71. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MA, Bender D, et al. PLINK: a tool set for whole-genome association and population-based linkage analyses. Am J Hum Genet. 2007;81(3):559–75.
72. Ha E, Bae SC, Kim K. Large-scale meta-analysis across East Asian and European populations updated genetic architecture and variant-driven biology of rheumatoid arthritis, identifying 11 novel susceptibility loci. Ann Rheum Dis. 2021;80(5):558–65.
73. Källberg H, Padyukov L, Plenge RM, Rönnelid J, Gregersen PK, van der Helm-van Mil AHM, et al. Gene-gene and gene-environment interactions involving HLA-DRB1, PTPN22, and smoking in two subsets of rheumatoid arthritis. Am J Human Genet. 2007;80(5):867–75.
74. Karlson EW, Deane K. Environmental and gene-environment interactions and risk of rheumatoid arthritis. Rheum Dis Clin. 2012;38(2):405–26.
75. Ishikawa Y, Terao C. The impact of cigarette smoking on risk of rheumatoid arthritis: a narrative review. Cells. 2020;9(2):475.
76. Speiser JL, Miller ME, Tooze J, Ip E. A comparison of random forest variable selection methods for classification prediction modeling. Expert Syst Appl. 2019;134:93–101.
77. Waldmann P, Mészáros G, Gredler B, Fürst C, Sölkner J. Evaluation of the lasso and the elastic net in genome-wide association studies. Front Genet. 2013;4:270.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Additional file 1

Evaluation of tree-based statistical learning methods for constructing genetic risk scores

Michael Lau[1,2,*], Claudia Wigmann[2], Sara Kress[2], Tamara Schikowski[2] and Holger Schwender[1]

[1] *Mathematical Institute, Heinrich Heine University, Düsseldorf, Germany*
[2] *IUF – Leibniz Research Institute for Environmental Medicine, Düsseldorf, Germany*
[*] *Correspondence: michael.lau@hhu.de*

In this supplementary file, additional information about the GRS construction methods and additional results about the simulation study and the real data application are presented. In Figure S1, model fitting and GRS prediction times are depicted. In Section 2, the considered hyperparameters for constructing the GRS models are described. In Section 3, we present the workflows for tuning and fitting each regarded statistical learning procedure for constructing GRS. Means and asymptotic 95% confidence intervals of the AUCs corresponding to the figures in the main text are depicted in the Figures S2, S9, and S16. Concrete estimates following statistical inference can be found in the Figures S3, S4, S10, S11, and in Table S1. Results for the classical classification metrics accuracy, sensitivity, and specificity are depicted in the Figures S5, S6, S7, S12, S13, S14, S17, S18, and S19. Training data AUCs are illustrated in the Figures S8, S15, S20, and S24. AUC comparisons when employing the binary $\{0, 1\}$ SNP coding for each method are depicted in the Figures S21, S22, and S23. Table S2 depicts median p-values of the final adjusted models for the GRS, the environmental factor, and their interaction term. Final results for the sensitivity analysis excluding smokers from the SALIA data set can be found in Figure S25. In Figure S26, an exemplary GRS distribution is depicted which explains the observed sensitivities in the simulation study.

1 Model fitting and GRS prediction time

We, here, present the model fitting and GRS prediction times in the third simulation scenario. The times for single model constructions and evaluations in the hyperparameter optimization process are presented, since, in the hyperparameter optimization process, several different settings, which can have an impact on the time, are utilized.

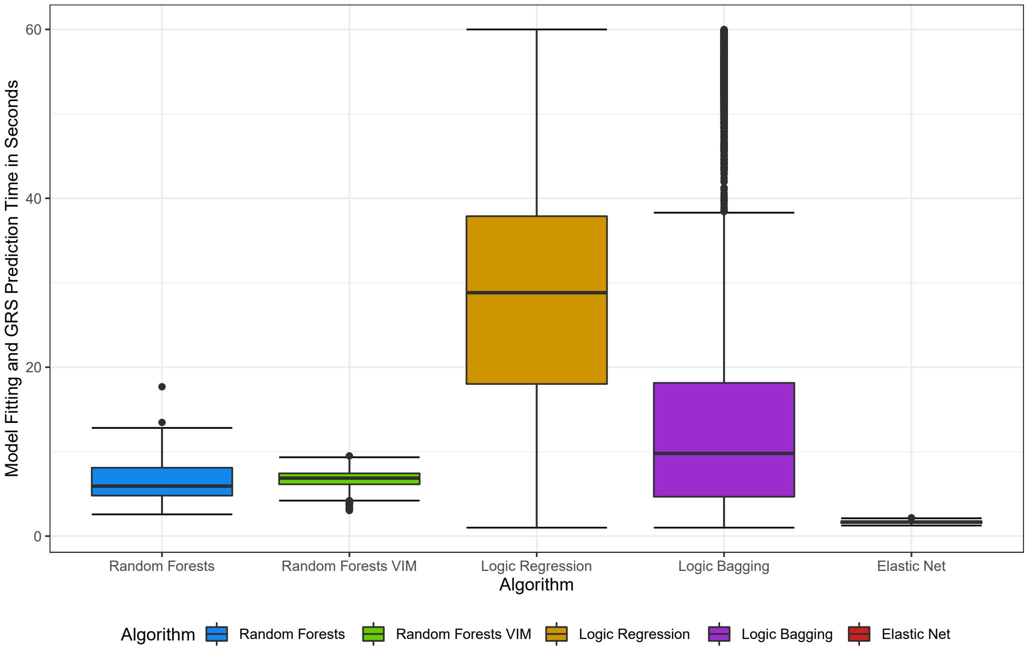

Figure S1: Model fitting and GRS prediction time for random forests, random forests VIM, logic regression, logic bagging, and elastic net for the hyperparameter configuration in the third simulation scenario incorporating continuous input variables.

2 Hyperparameter descriptions

We, here, briefly describe the hyperparameters of each considered statistical learning procedure that were tuned in our analyses. Table 4 in the main text depicts the corresponding hyperparameter settings.

2.1 Random forests & random forests VIM

The parameter mtry determines the number of randomly chosen input variables regarded at each split in each tree. The parameter min.node.size configures the number of observations which have to belong to a certain tree node in order to continue splitting this node. Thus, min.node.size acts as a stopping criterion for prematurely terminating splitting of a tree branch. num.trees determines the total number of trees to be grown in random forests. A sufficiently high number should be chosen such that the performance will not increase substantially anymore.

2.2 Logic regression & logic bagging

For logic regression and logic bagging, ntrees and nleaves determine the model complexity. ntrees is the maximum number of trees to be included in the model and nleaves is the maximum number of leaves distributed over all trees.

For conventional logic regression, simulated annealing is employed as the search algorithm which has to be tuned as well. For the number of simulated annealing iterations, analogously to the number of trees in random forests, a sufficiently high number should be chosen. The cooling schedule, which includes a start temperature and an end temperature, is manually tuned such that at the beginning of the search, almost all states are accepted, and at the end of the search, almost no states are accepted.

For logic bagging, the number of bagging iterations has to be set to a sufficiently high number, similar to num.trees and the number of simulated annealing iterations.

2.3 Elastic net

For fitting elastic net models, the parameter α controls the balance between the lasso and the ridge regularization. The parameter λ determines the strength of the regularization.

3 Tuning and training workflows

Since each statistical learning method regarded in this article requires considering different details for properly fitting GRS models, we here briefly present the workflows for each method.

3.1 Random forests

1. Choose a sufficiently high number of trees to be fitted, e.g., 2000

2. Tune the minimum node size and the number of randomly chosen predictors at each split in each tree using a grid search by fitting a random forest with probability estimation trees for each eligible setting

3. Fit a random forest with probability estimation trees using the best identified hyperparameter configuration

3.2 Random forests VIM

1. Choose a sufficiently high number of trees to be fitted, e.g., 2000

2. Tune the minimum node size and the number of randomly chosen predictors at each split in each tree using a grid search by performing a variable selection via the Boruta approach and fitting a random forest with probability estimation trees for each eligible setting

3. Perform a variable selection via the Boruta approach and fit a random forest with probability estimation trees using the best identified hyperparameter configuration

3.3 Logic regression

1. Split all considered SNPs into two binary variables coding for dominant and recessive effects

2. Choose a sufficiently high number of markov chain iterations to be executed, e.g., 500000

3. Experimentally tune the cooling schedule for simulated annealing, i.e., choose a start temperature such that almost all states are accepted and choose a final temperature such that almost no states are accepted

4. Tune the number of trees and the total number of leaves using a grid search by fitting a logic regression model with the logit link function for each eligible setting

5. Fit a logic regression model with the logit link function using the best identified hyperparameter configuration

3.4 Logic bagging

1. Split all considered SNPs into two binary variables coding for dominant and recessive effects

2. Choose a sufficiently high number of bagging iterations to be performed, e.g., 500

3. Tune the number of trees and the total number of leaves using a grid search by fitting a logic bagging model with the logit link function for each eligible setting. A logic bagging model is fitted by drawing a bootstrap sample and fitting a logic regression model with a greedy search to this sample for each bagging iteration.

4. Fit a logic bagging model with the logit link function using the best identified hyperparameter configuration

3.5 Elastic net

1. Tune the elastic net parameter α using a grid search by fitting an elastic net model with the logit link function for each eligible setting. Automatically configure the regularization parameter λ by performing an inner cross-validation (`cv.glmnet` in `glmnet`).

2. Fit an elastic net model with the logit link function using the best identified hyperparameter configuration

4 Simulation studies

4.1 Marginal genetic effects

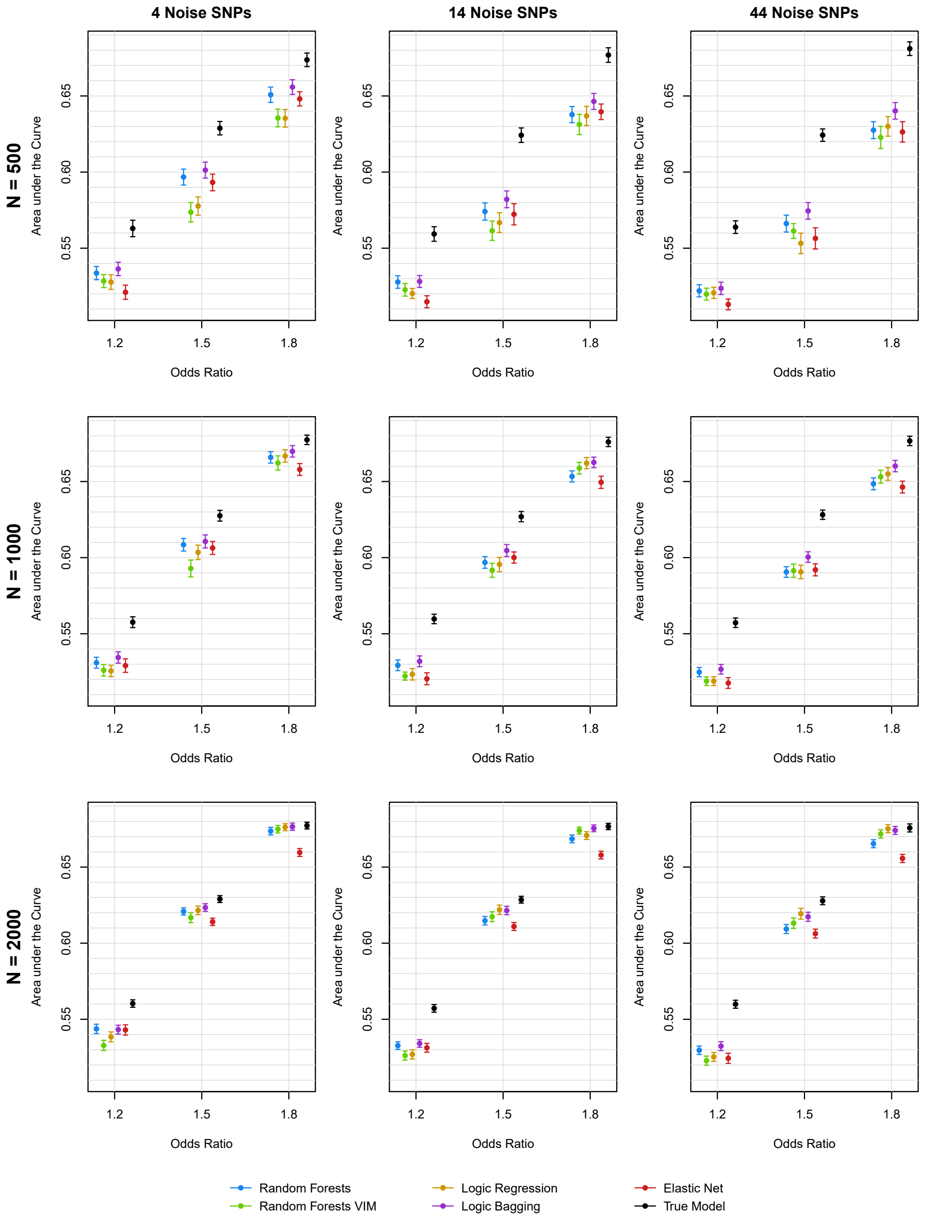

Figure S2: Mean AUC and asymptotic 95% confidence intervals for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the first simulation scenario considering marginal effective SNPs evaluated on the test data.

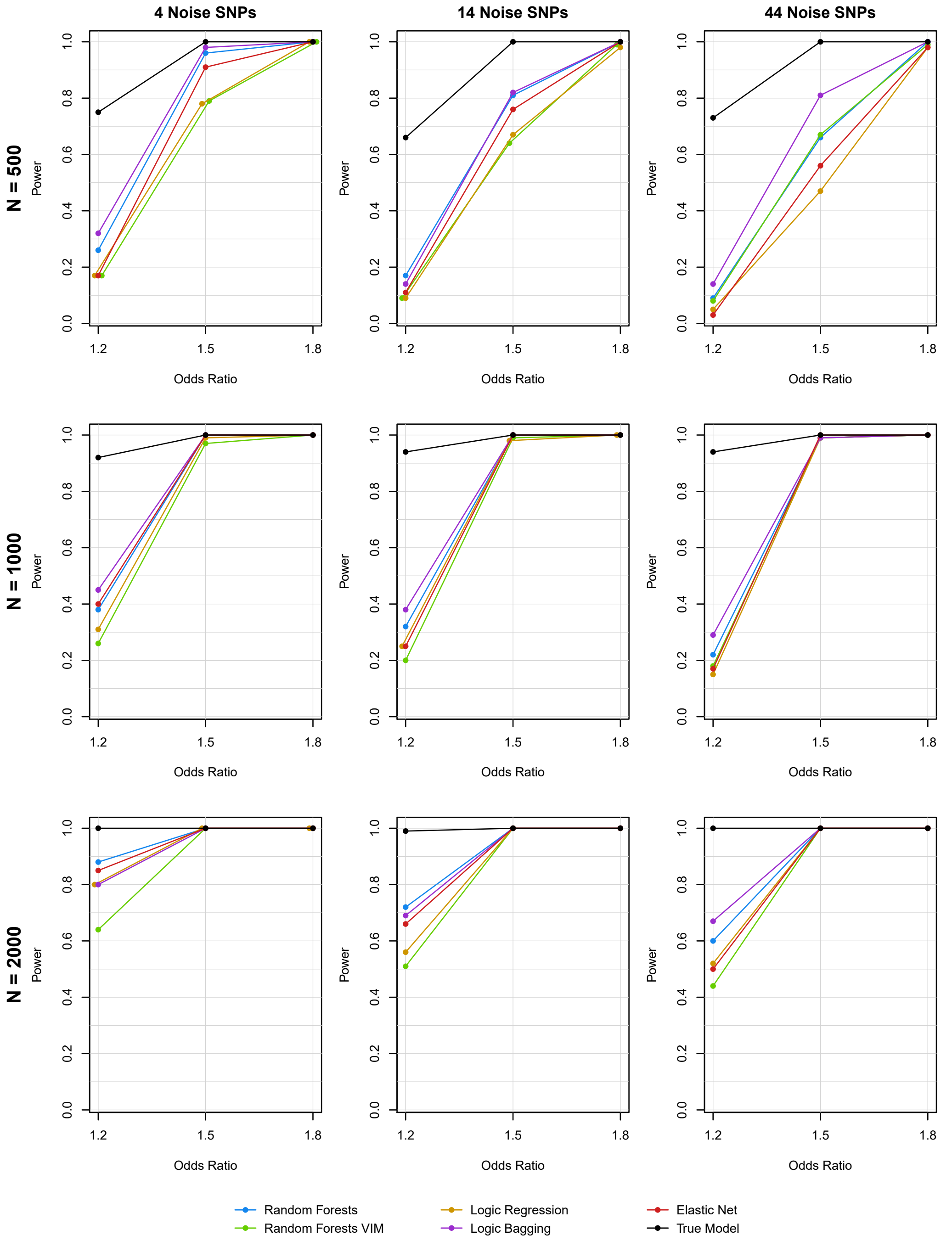

Figure S3: Estimated power for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the first simulation scenario considering marginal effective SNPs evaluated on the test data.

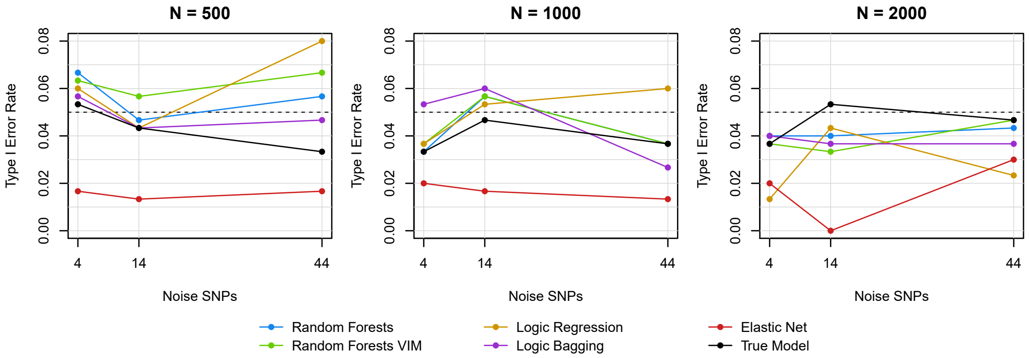

Figure S4: Estimated type I error rate for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the first simulation scenario considering marginal effective SNPs evaluated on the test data.

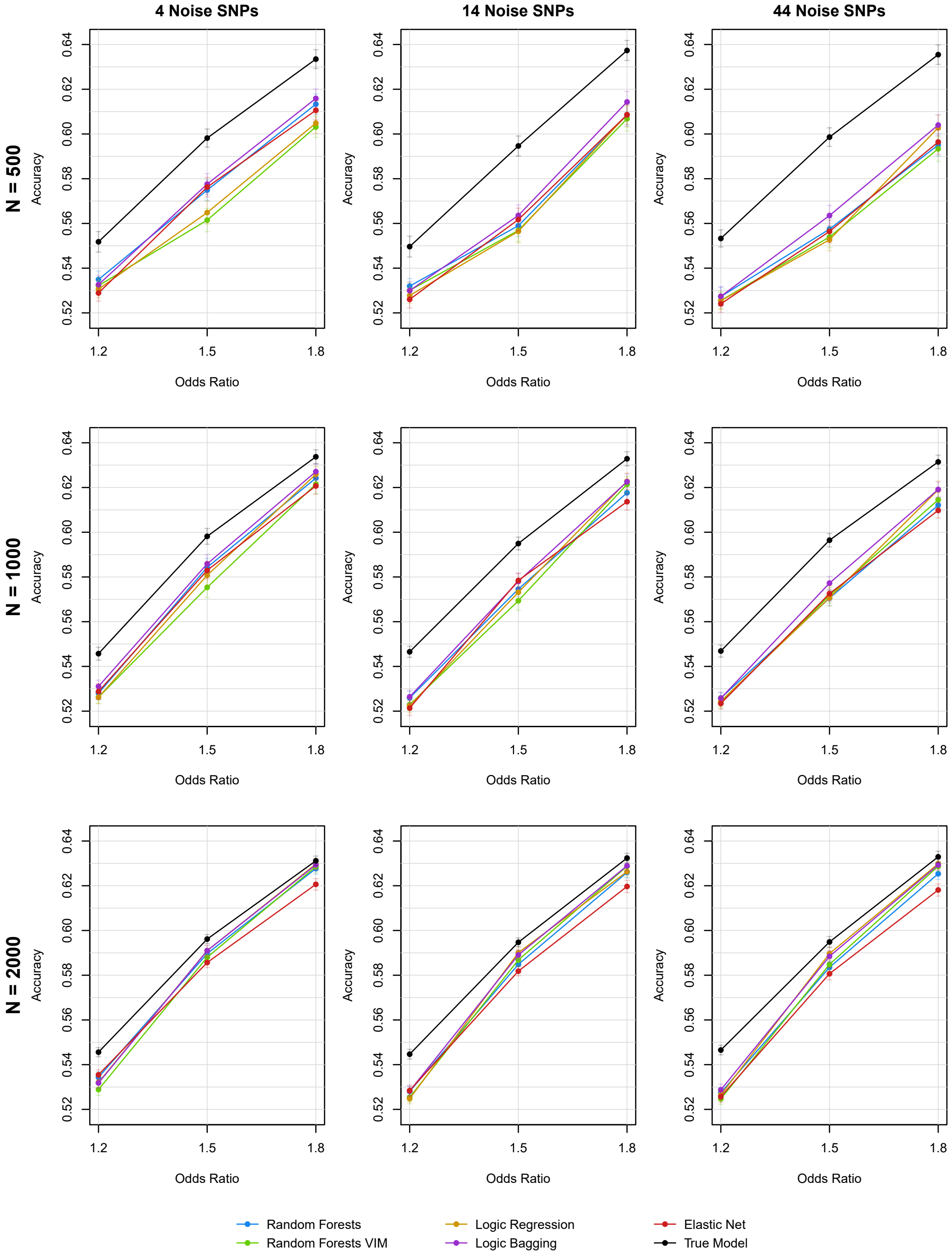

Figure S5: Mean accuracy for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the first simulation scenario considering marginal effective SNPs evaluated on the test data.

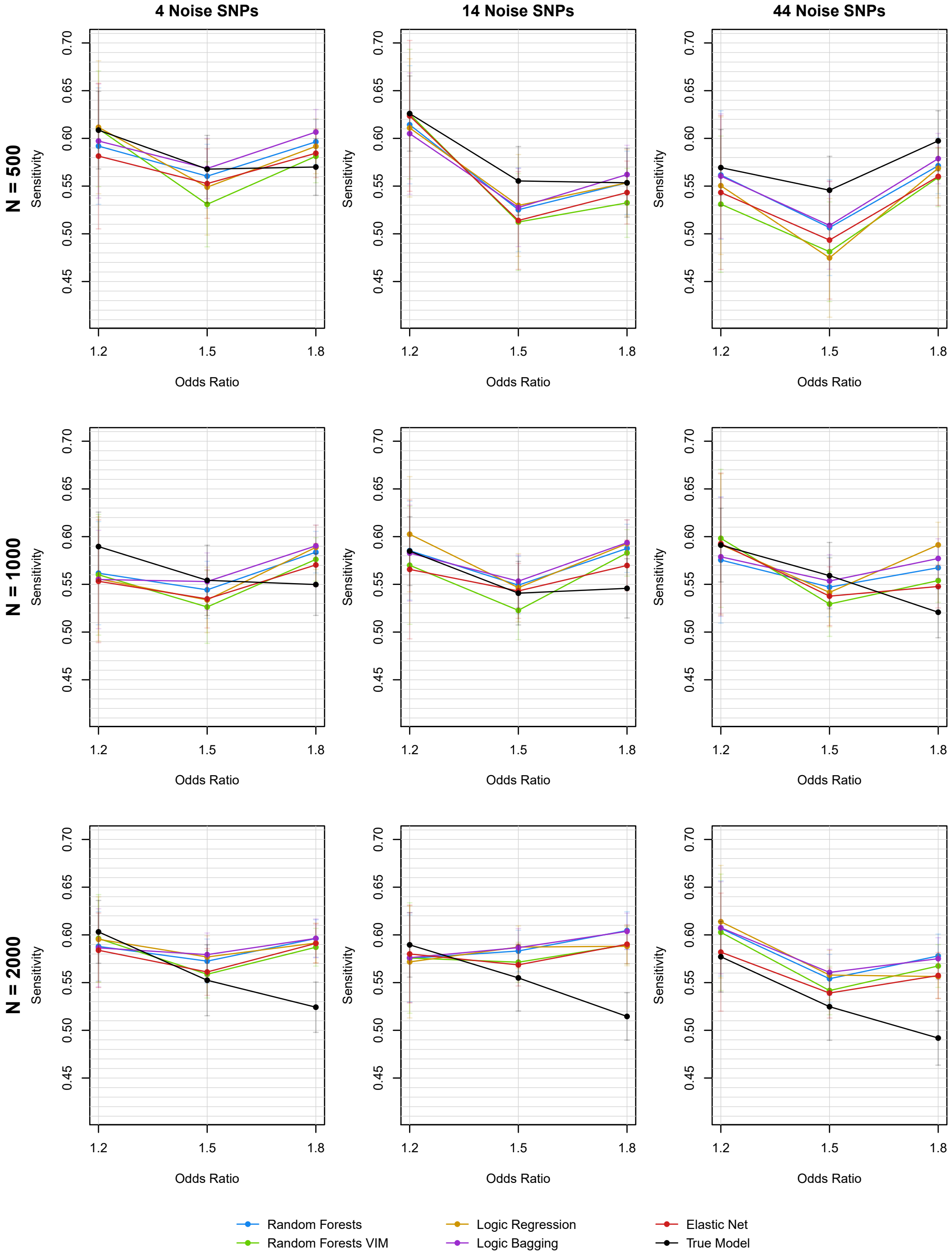

Figure S6: Mean sensitivity for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the first simulation scenario considering marginal effective SNPs evaluated on the test data.

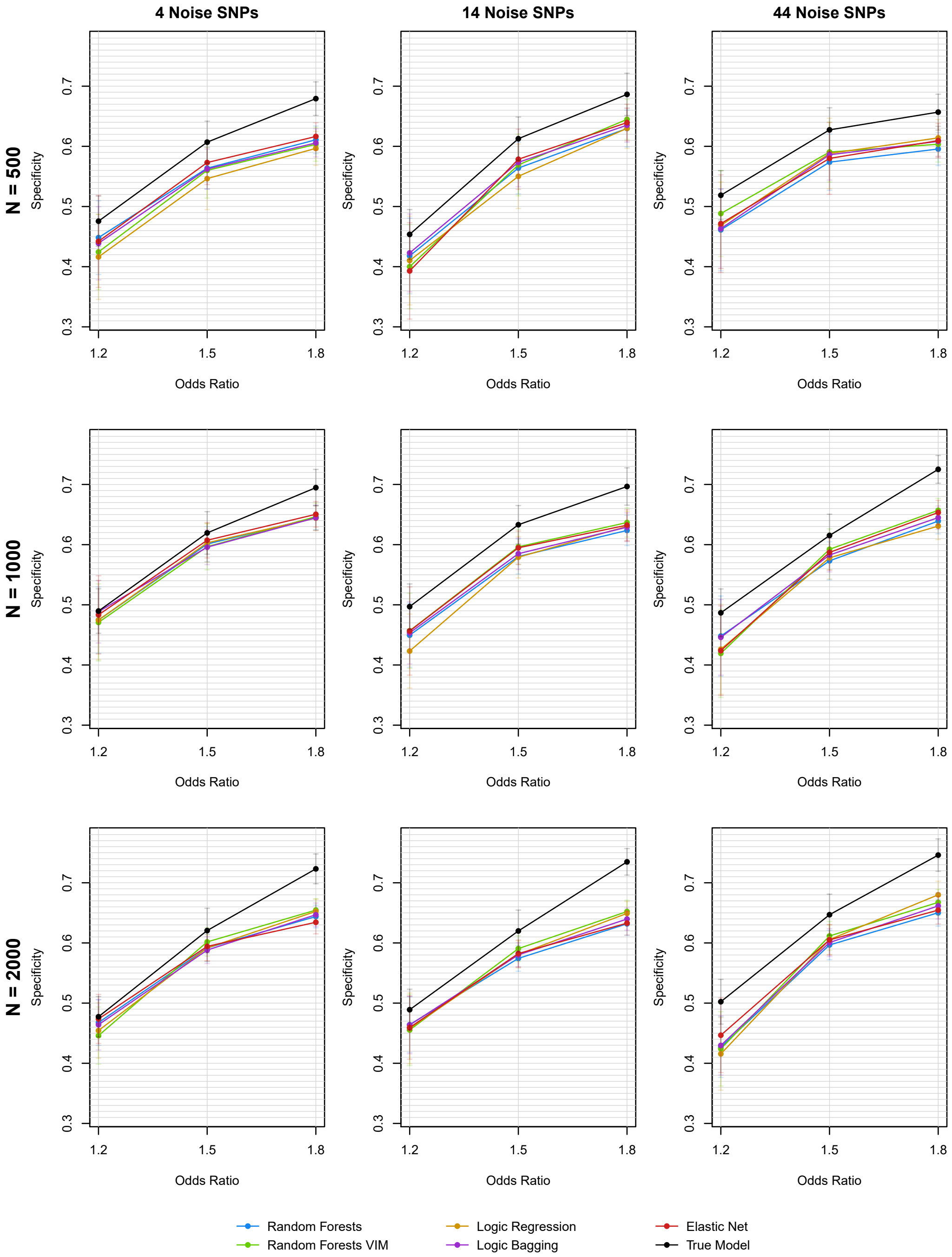

Figure S7: Mean specificity for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the first simulation scenario considering marginal effective SNPs evaluated on the test data.

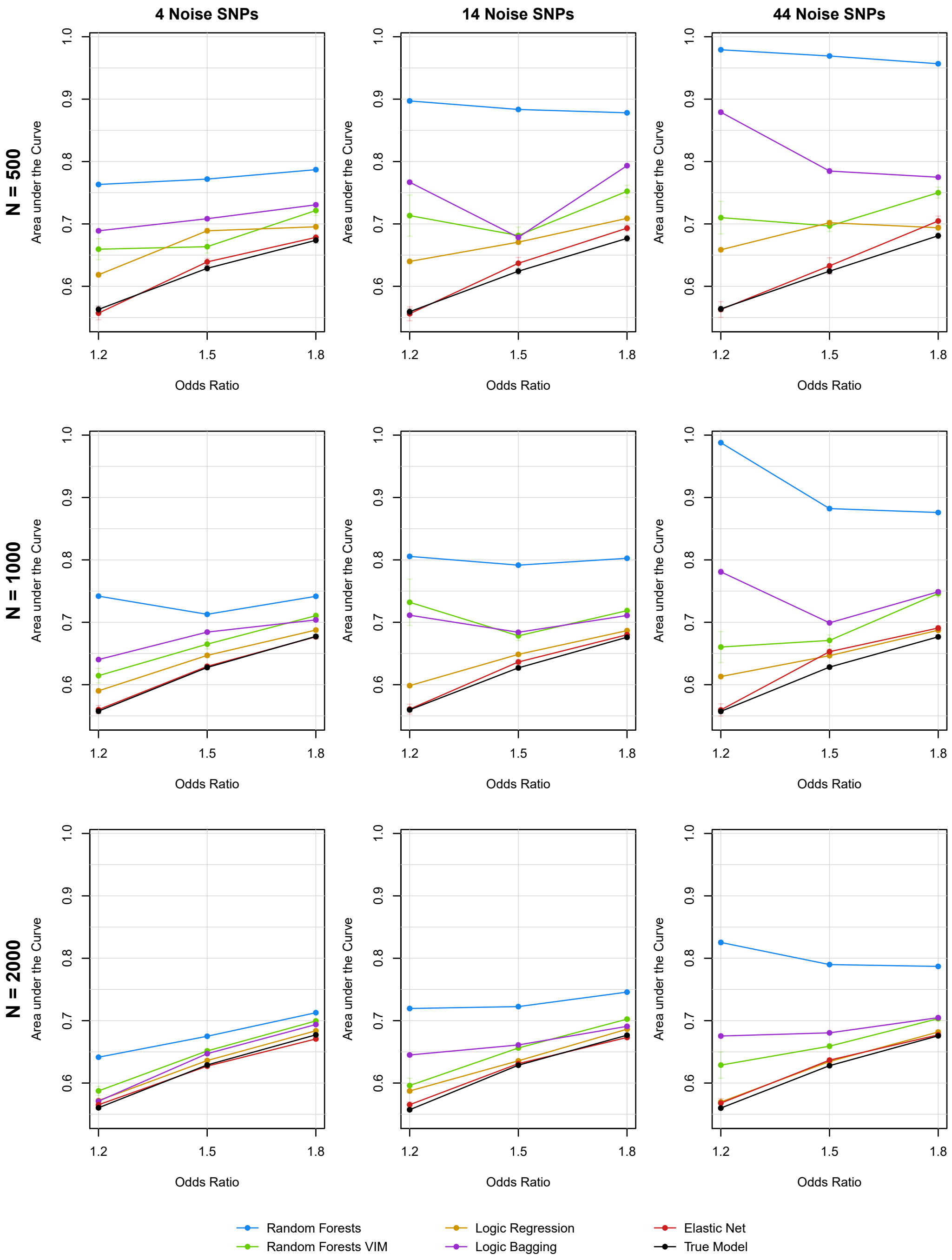

Figure S8: Mean AUC for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the first simulation scenario considering marginal effective SNPs evaluated on the training data itself.

4.2 Dominant interaction effects of SNPs

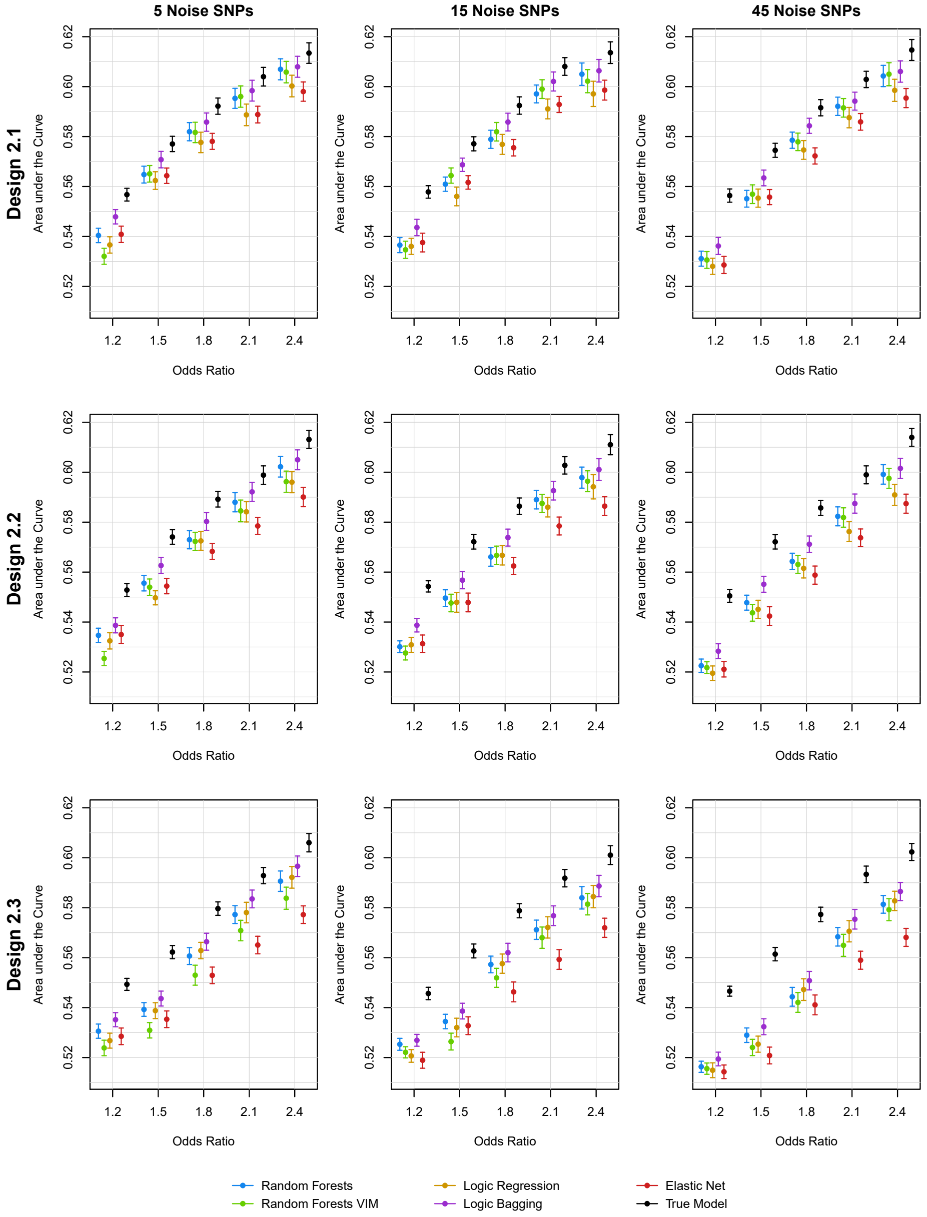

Figure S9: Mean AUC and asymptotic 95% confidence intervals for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the second simulation scenario incorporating interactions of SNPs evaluated on the test data.

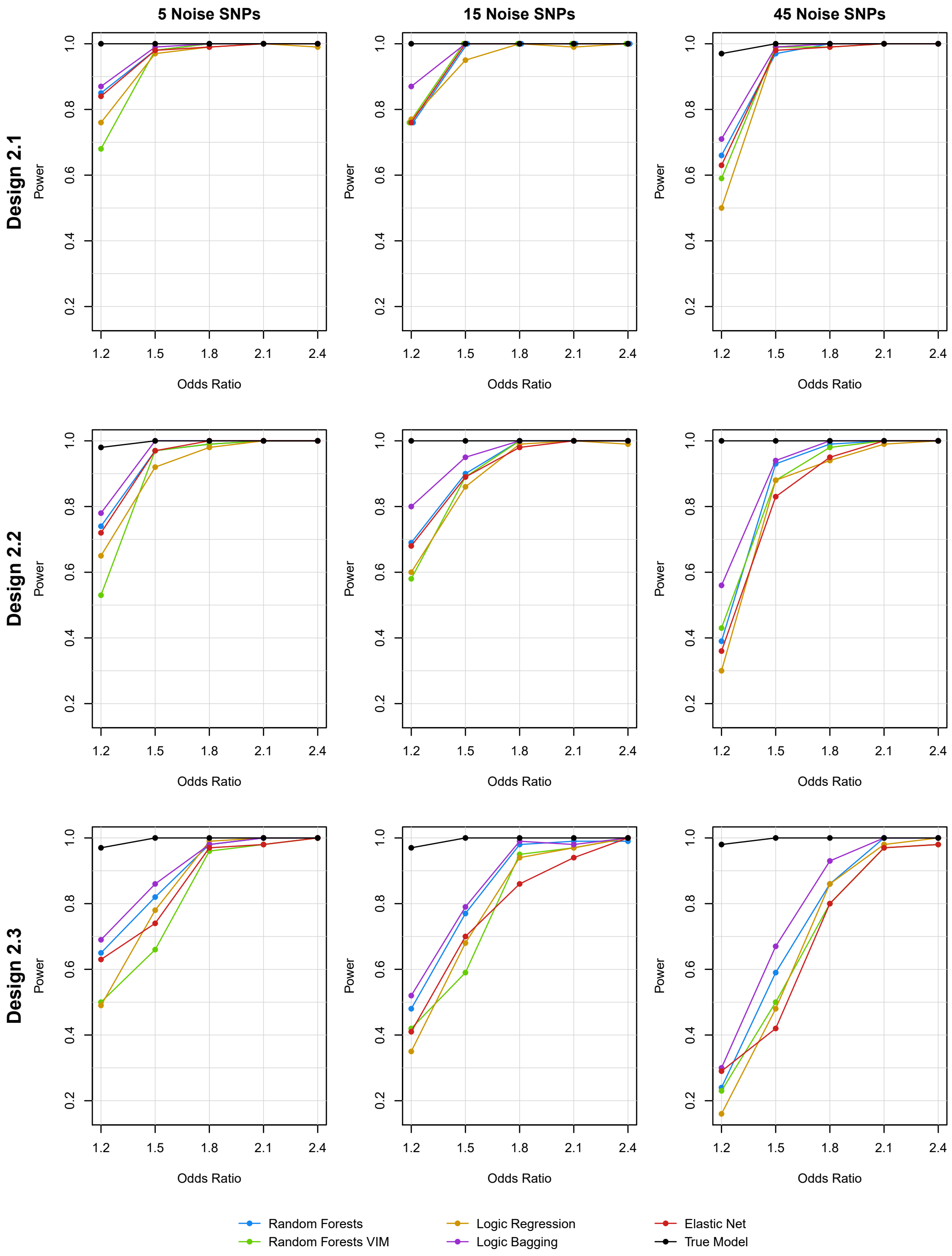

Figure S10: Estimated power for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the second simulation scenario incorporating interactions of SNPs evaluated on the test data.

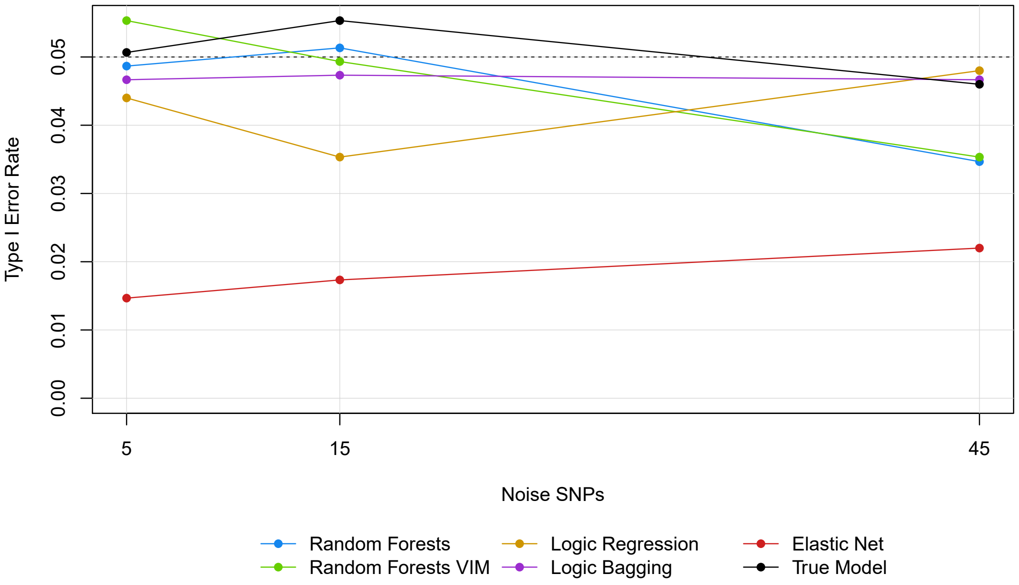

Figure S11: Estimated type I error rate for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the second simulation scenario incorporating interactions of SNPs evaluated on the test data.

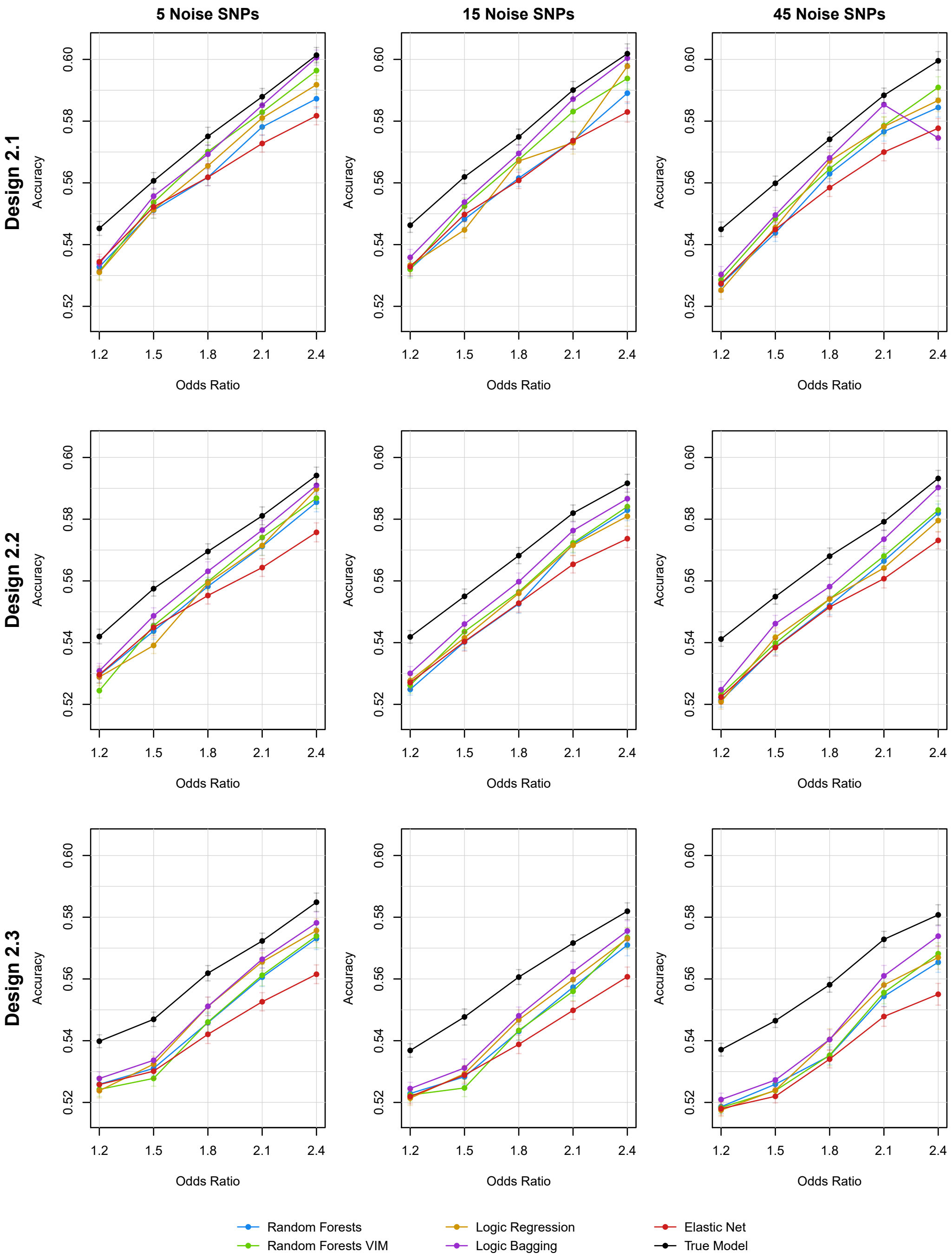

Figure S12: Mean accuracy for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the second simulation scenario incorporating interactions of SNPs evaluated on the test data.

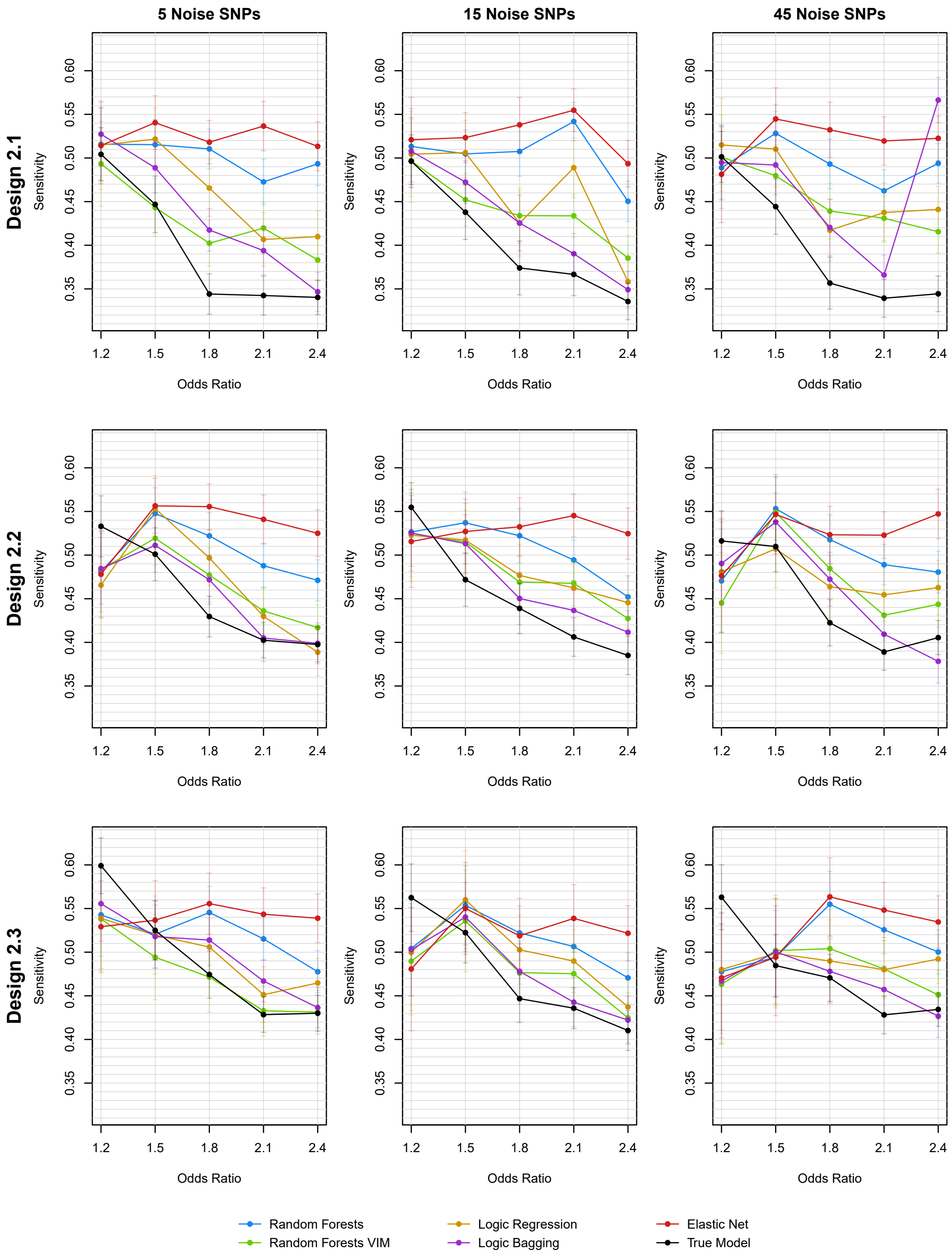

Figure S13: Mean sensitivity for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the second simulation scenario incorporating interactions of SNPs evaluated on the test data.

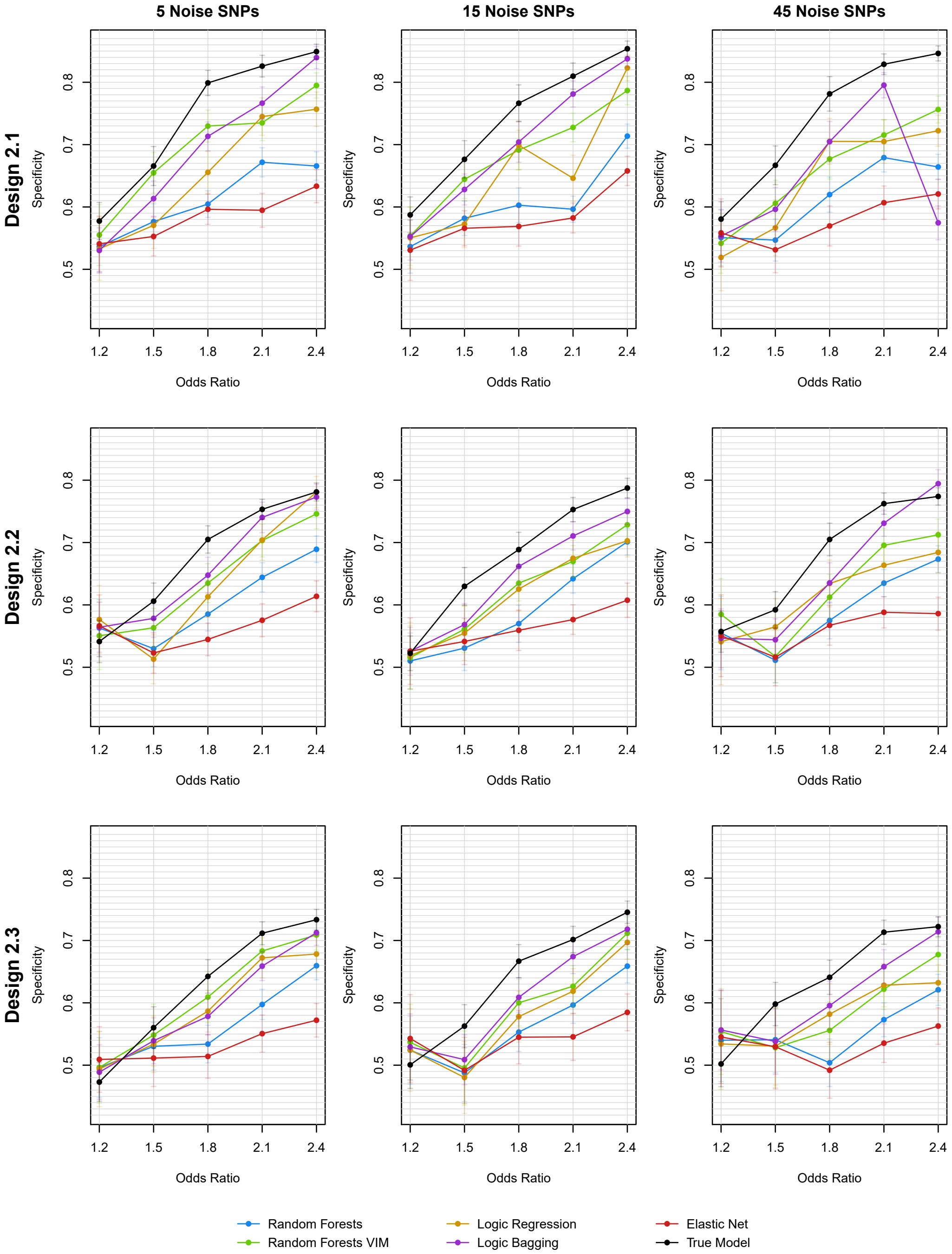

Figure S14: Mean specificity for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the second simulation scenario incorporating interactions of SNPs evaluated on the test data.

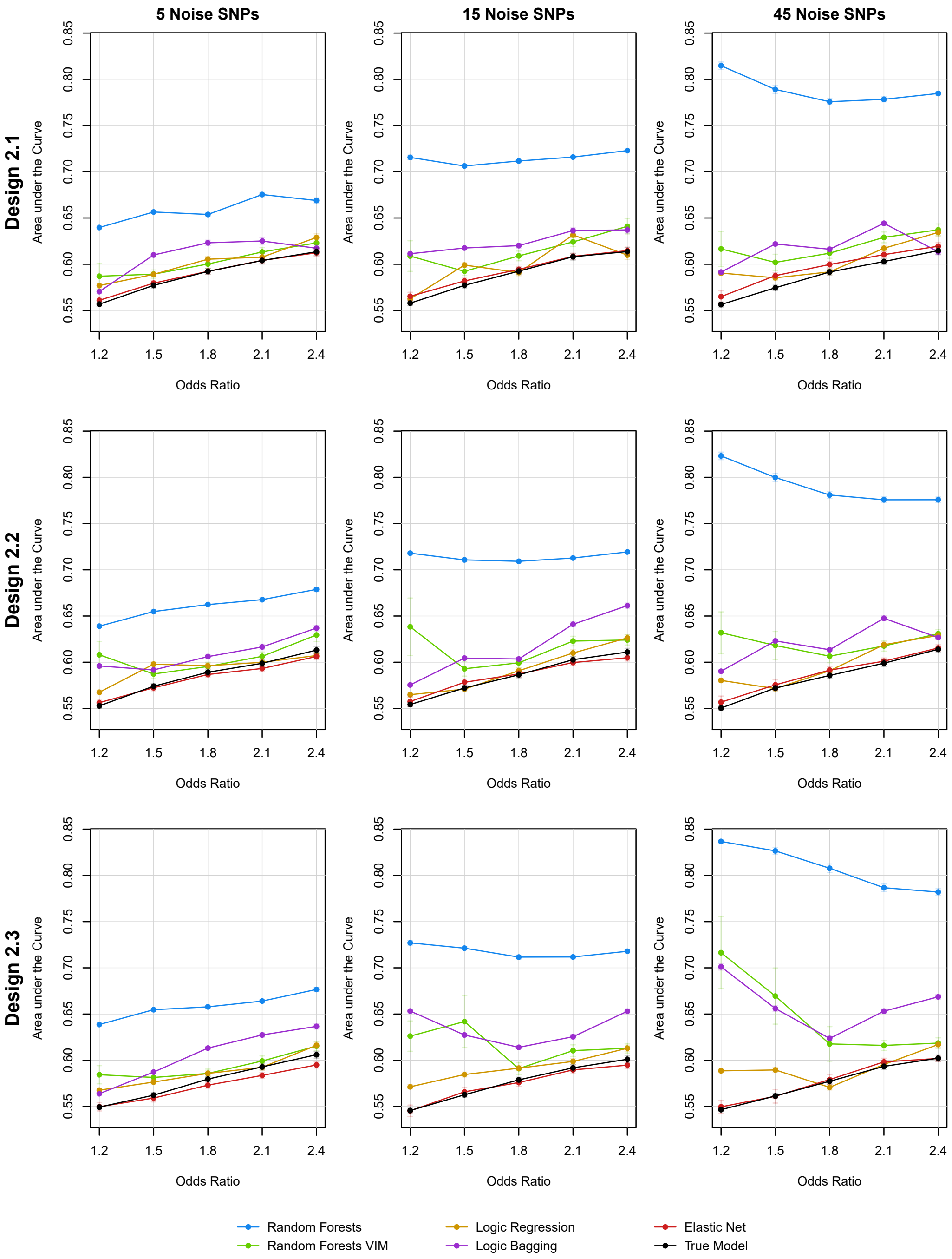

Figure S15: Mean AUC for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the second simulation scenario incorporating interactions of SNPs evaluated on the training data itself.

4.3 Gene-environment interactions

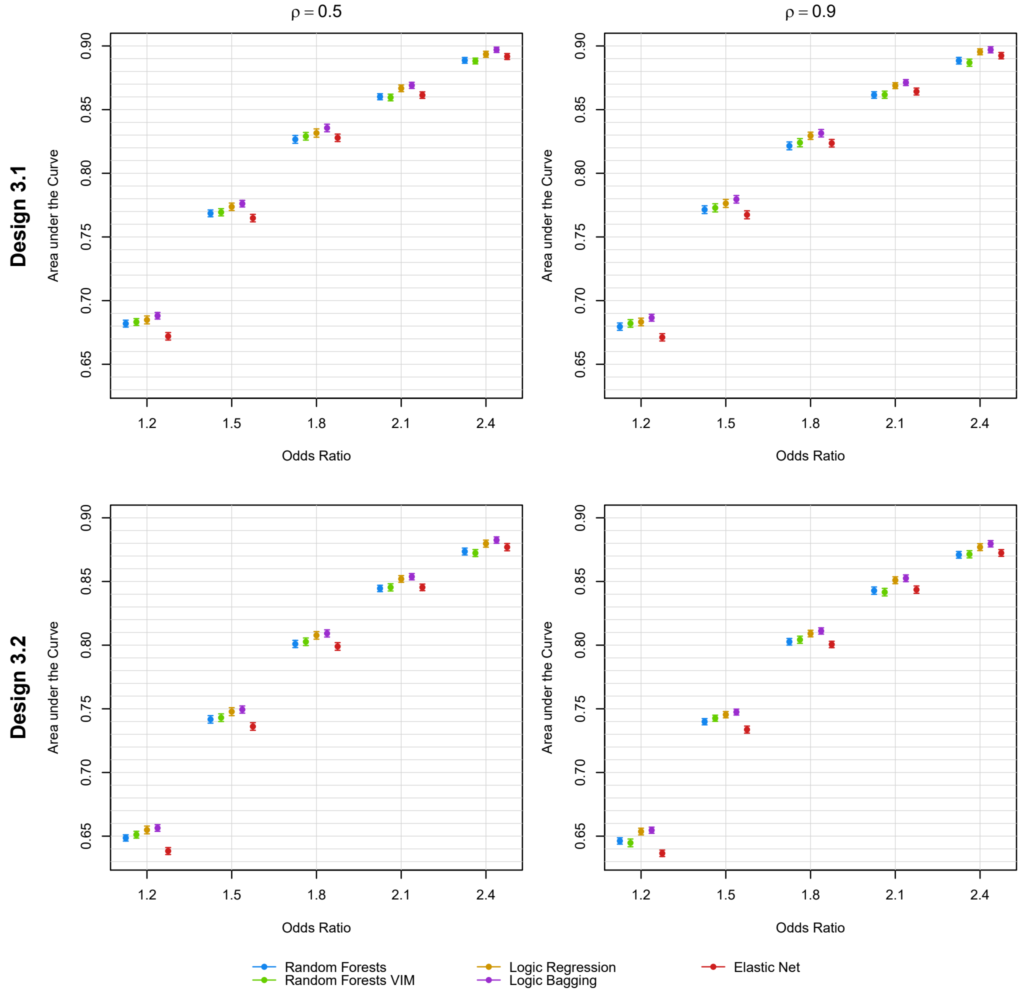

Figure S16: Mean AUC and asymptotic 95% confidence intervals for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the third simulation scenario incorporating continuous input variables evaluated on the test data.

Table S1: Estimated type I error rate for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the third simulation scenario incorporating continuous input variables evaluated on the test data.

Algorithm	Type I Error Rate
Random Forests	0.056
Random Forests VIM	0.052
Logic Regression	0.051
Logic Bagging	0.054
Elastic Net	0.020

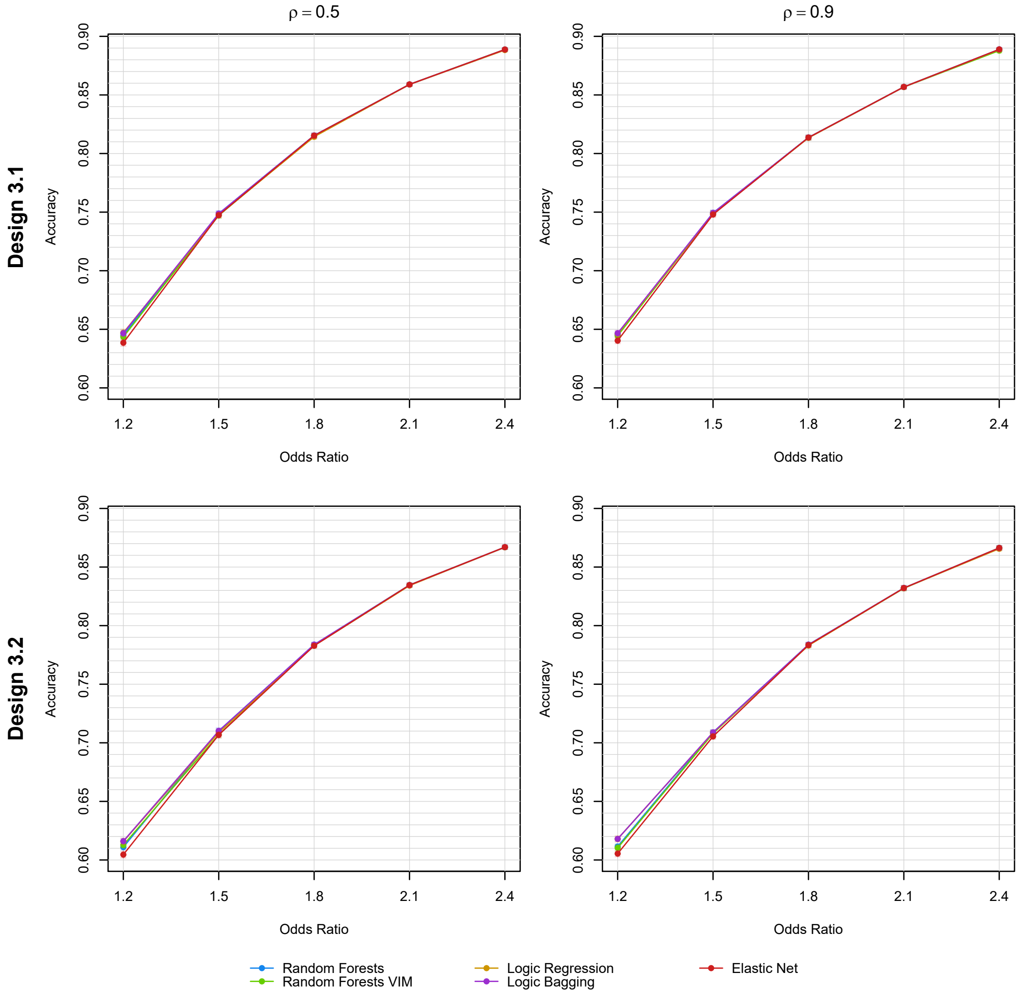

Figure S17: Mean accuracy for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the third simulation scenario incorporating continuous input variables evaluated on the test data.

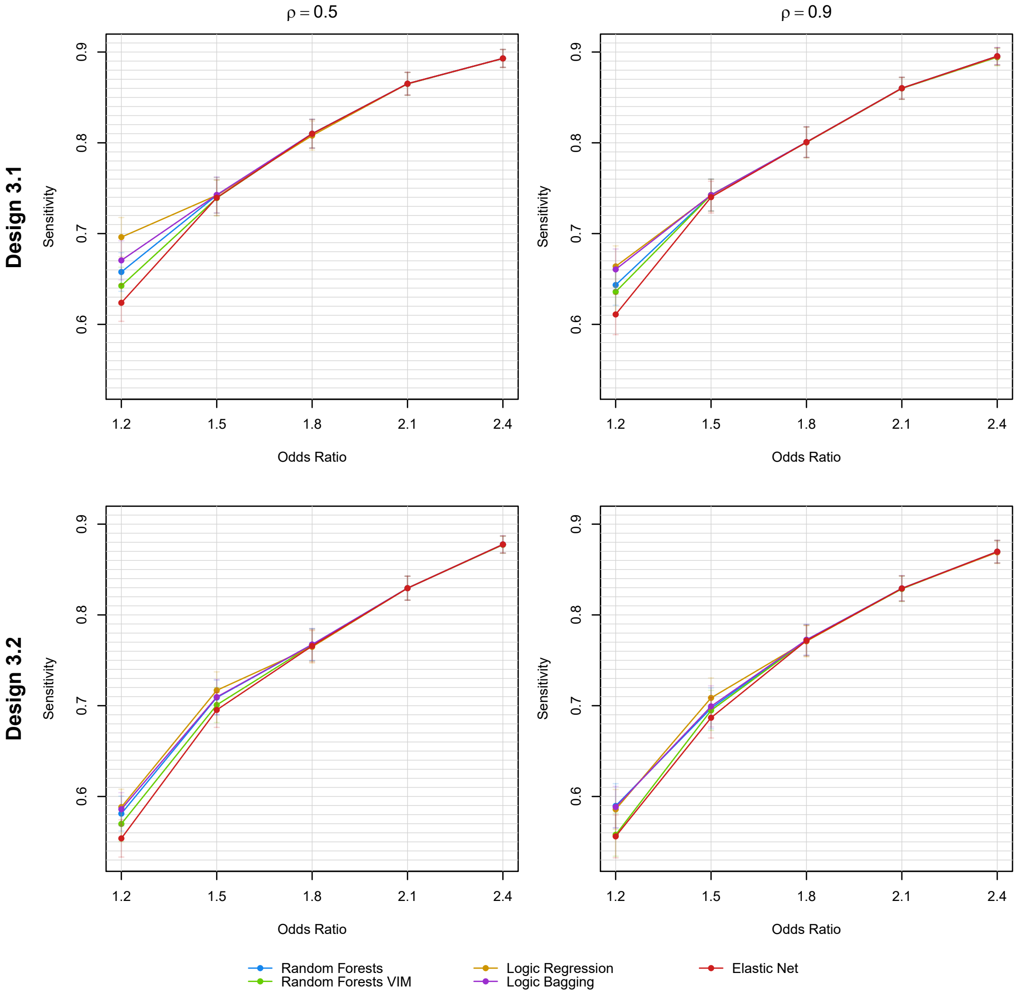

Figure S18: Mean sensitivity for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the third simulation scenario incorporating continuous input variables evaluated on the test data.

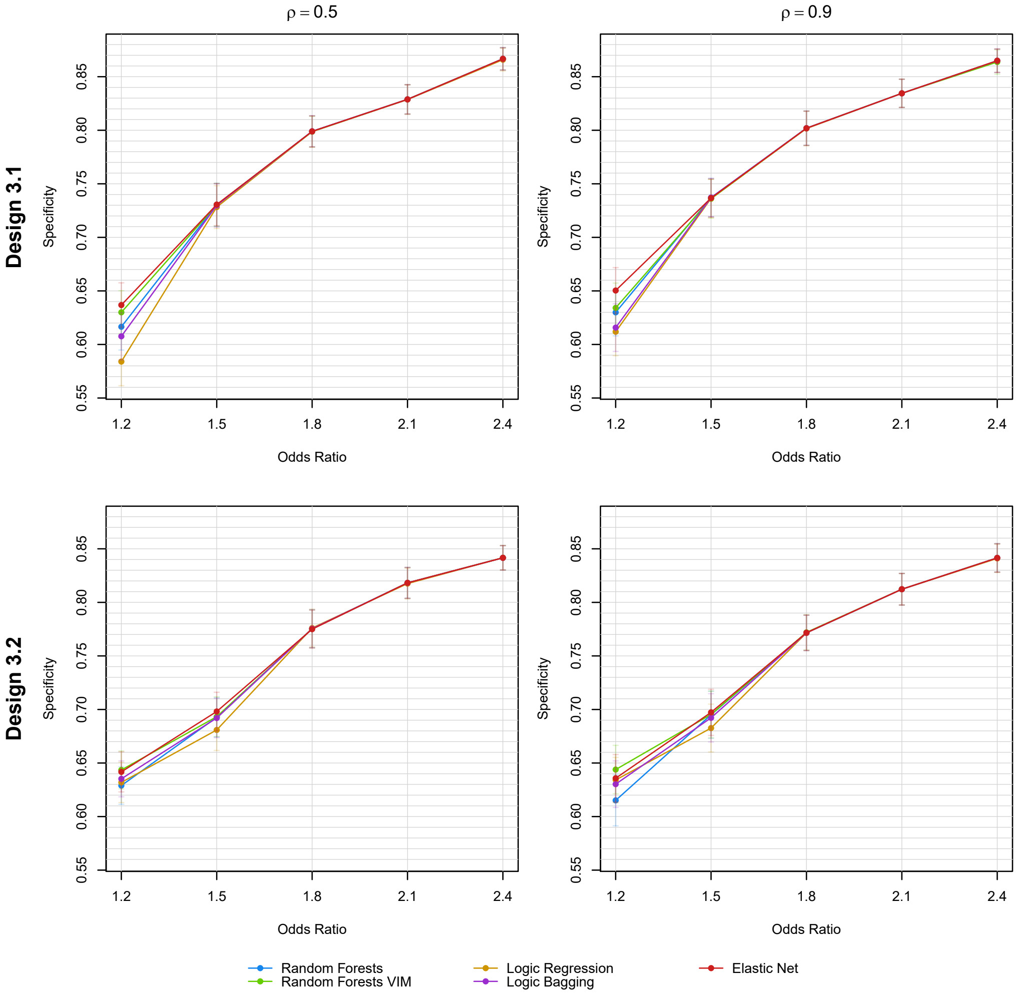

Figure S19: Mean specificity for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the third simulation scenario incorporating continuous input variables evaluated on the test data.

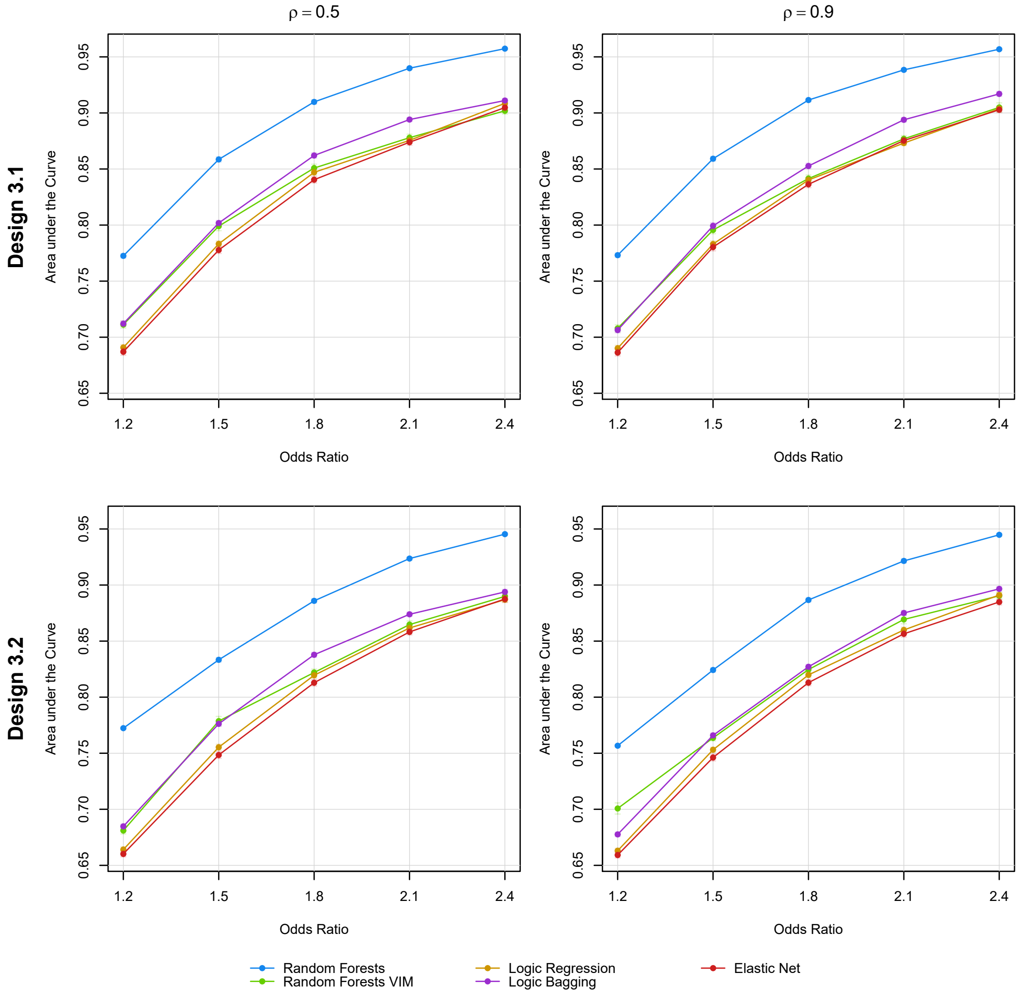

Figure S20: Mean AUC for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the third simulation scenario incorporating continuous input variables evaluated on the training data itself.

4.4 Comparison considering binary SNP codings

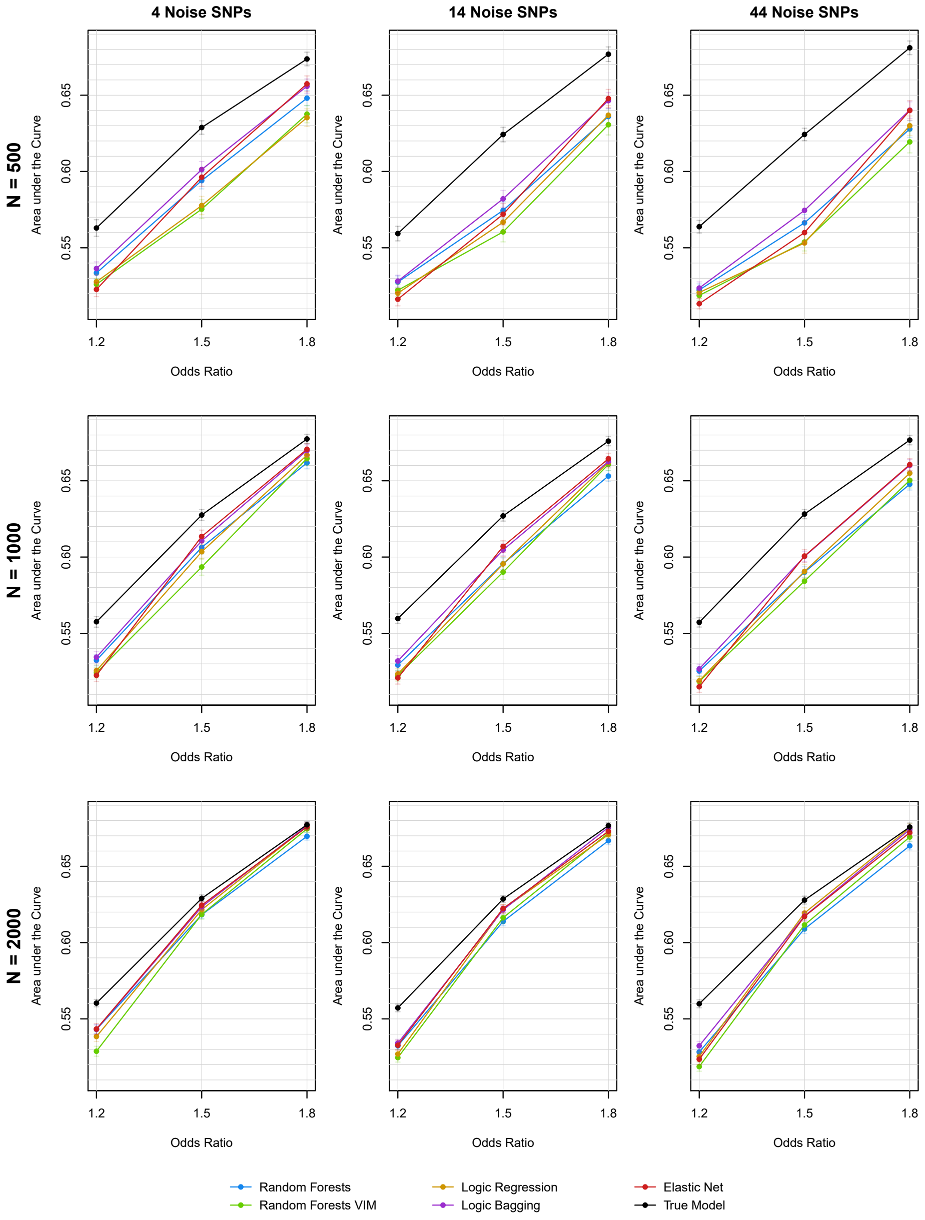

Figure S21: Mean AUC for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the first simulation scenario considering marginal effective SNPs evaluated on the test data. Here, the binary $\{0, 1\}$ SNP coding was used for each method.

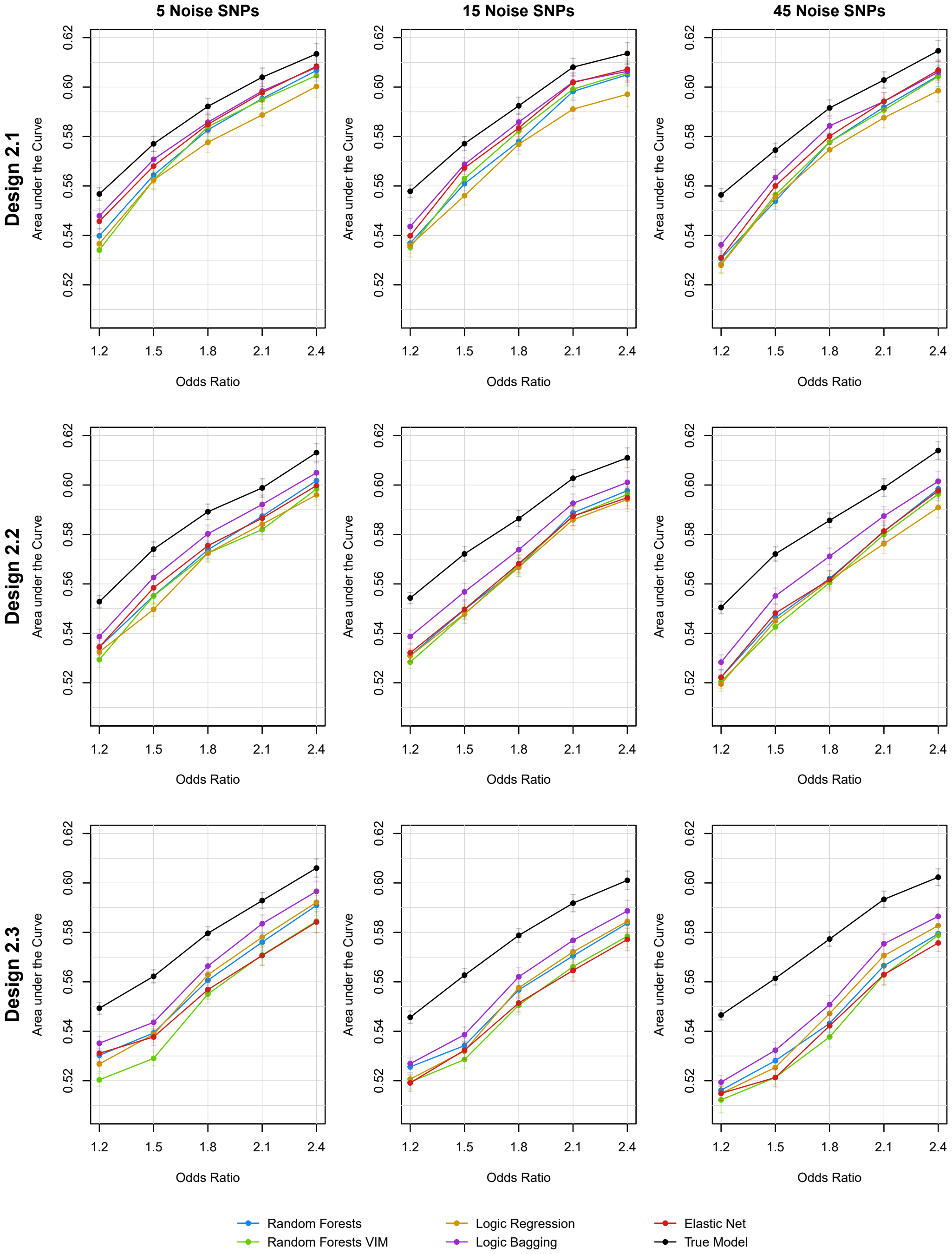

Figure S22: Mean AUC for random forests, random forests VIM, logic regression, logic bagging, elastic net, and the true underlying model in the second simulation scenario incorporating interactions of SNPs evaluated on the test data. Here, the binary $\{0, 1\}$ SNP coding was used for each method.

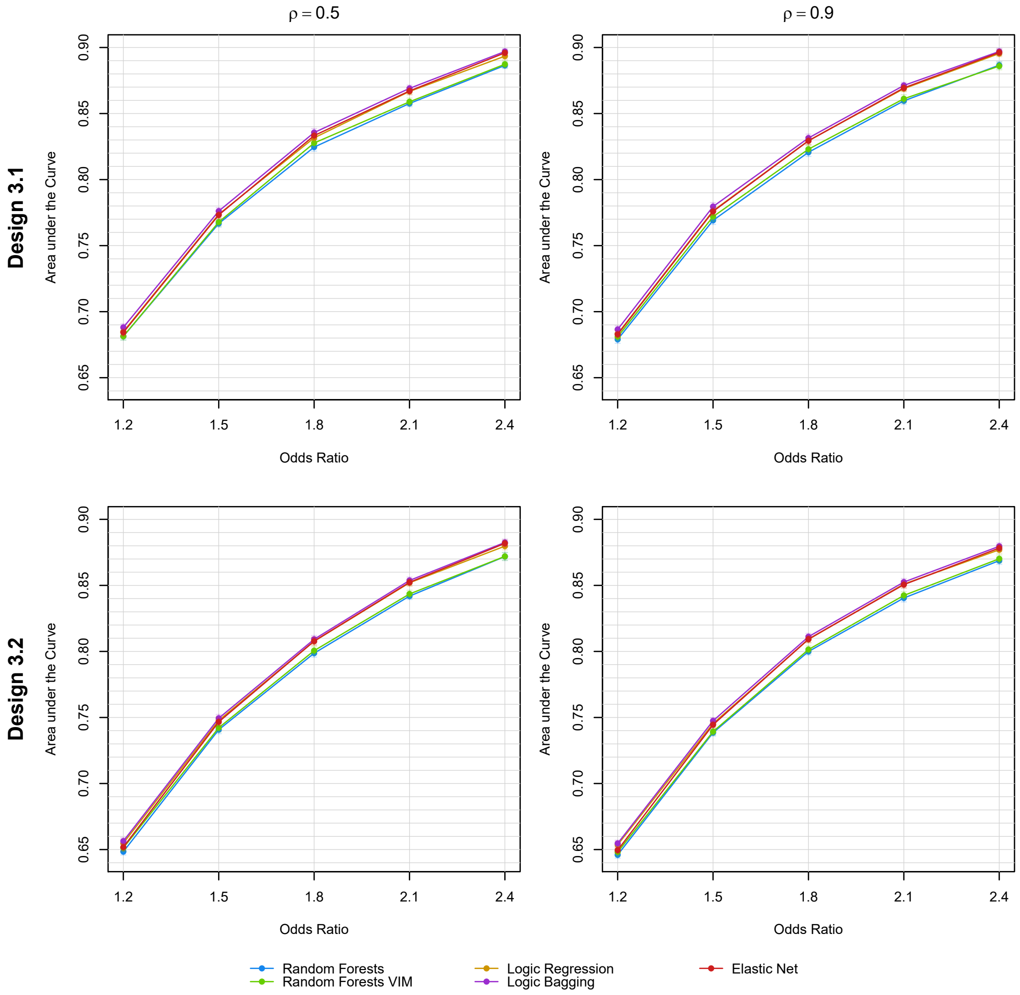

Figure S23: Mean AUC for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the third simulation scenario incorporating continuous input variables evaluated on the test data. Here, the binary $\{0, 1\}$ SNP coding was used for each method.

5 Real data application

Table S2: Median p-values of the Wald tests for the final age-adjusted models built on the SALIA data set

Term	Algorithm	NO_2	NO_x	PM_{10}	PM_{coarse}	$PM_{2.5}$	$PM_{2.5\ absorbance}$
GRS	Random Forests	0.469	0.385	0.531	0.550	0.332	0.539
	Random Forests VIM	0.485	0.432	0.416	0.470	0.404	0.449
	Logic Regression	0.430	0.420	0.394	0.452	0.338	0.400
	Logic Bagging	0.427	0.368	0.463	0.502	0.228	0.492
	Elastic Net	0.701	0.691	0.690	0.705	0.787	0.678
E	Random Forests	0.377	0.417	0.493	0.505	0.535	0.330
	Random Forests VIM	0.432	0.432	0.501	0.444	0.489	0.296
	Logic Regression	0.243	0.273	0.267	0.330	0.235	0.125
	Logic Bagging	0.378	0.388	0.485	0.539	0.513	0.249
	Elastic Net	0.304	0.356	0.425	0.333	0.421	0.250
GxE	Random Forests	0.489	0.538	0.575	0.591	0.511	0.530
	Random Forests VIM	0.505	0.402	0.401	0.460	0.457	0.490
	Logic Regression	0.467	0.404	0.417	0.432	0.440	0.407
	Logic Bagging	0.563	0.511	0.512	0.575	0.444	0.480
	Elastic Net	0.775	0.780	0.742	0.795	0.748	0.666

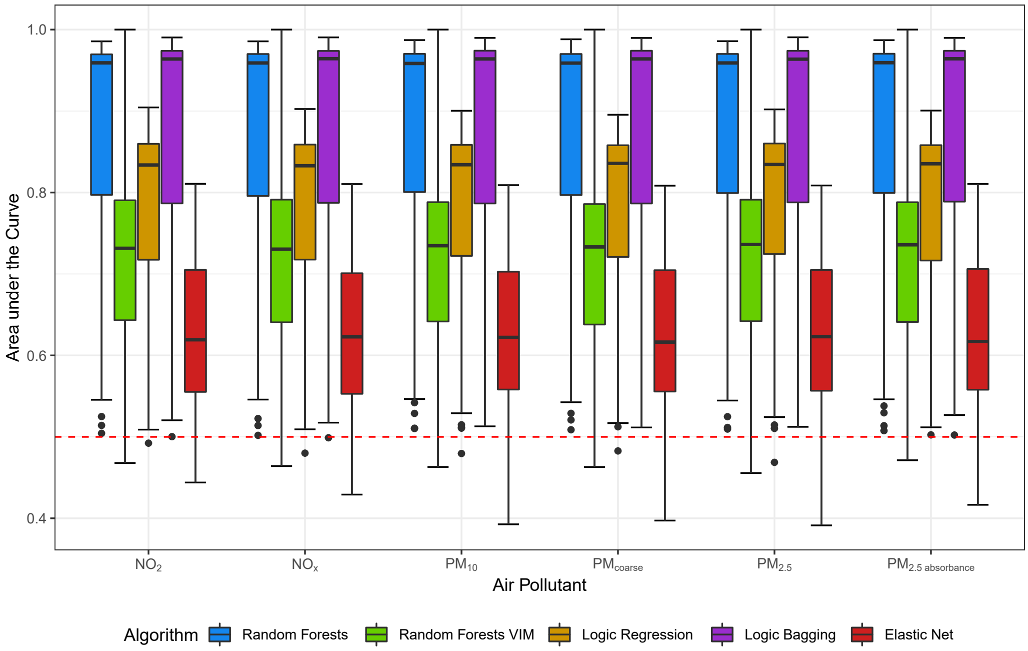

Figure S24: AUC for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the application to data from the SALIA study evaluated on the training data itself. Results for the final age-adjusted models with different air pollution indicators.

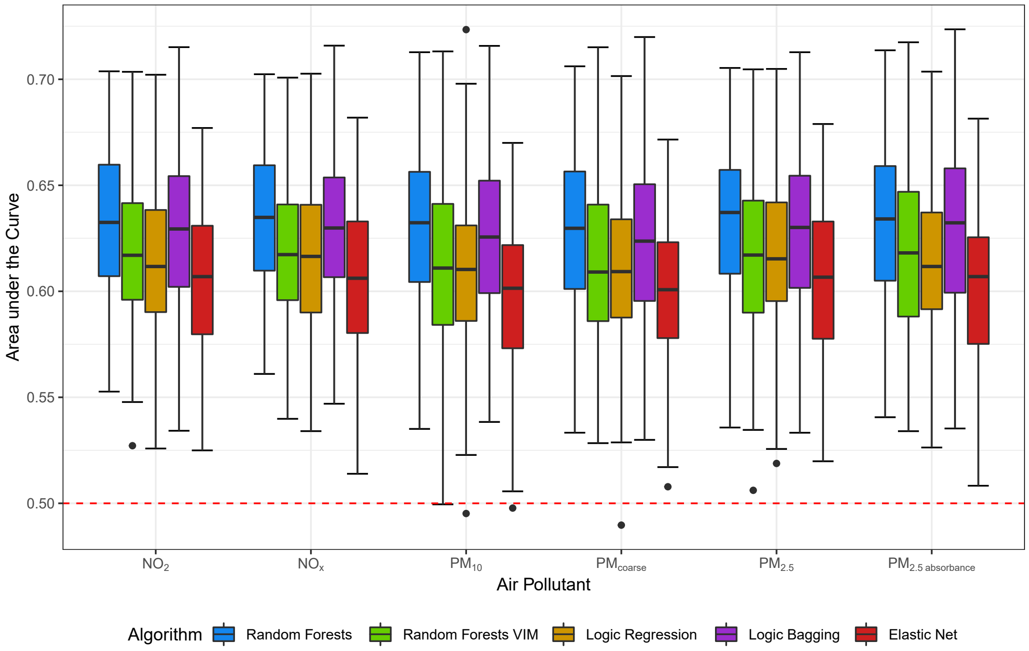

Figure S25: AUC for random forests, random forests VIM, logic regression, logic bagging, and elastic net in the application to data from the SALIA study evaluated on the test data. Results for the final age-adjusted models with different air pollution indicators. Current and former smokers were excluded from the base data set as part of a sensitivity analysis.

6 Distribution of the GRS

In the main effects simulation scenario and in the gene-gene interaction effect simulation scenario, the classification sensitivity is relatively low in some settings. This phenomenon can be explained by the need of dichotomizing the GRS into cases and controls for estimating the sensitivity and the discrete structure of the space of input variables/SNPs. To illustrate this, we present an exemplary GRS distribution occurring in the simulation study.

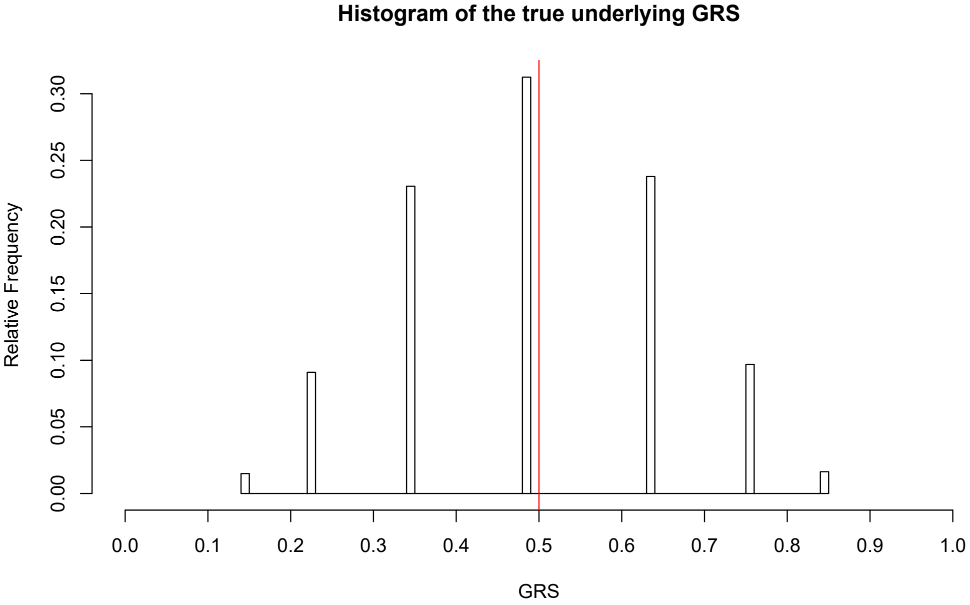

Figure S26: Histogram of the true underlying GRS for the main effects simulations scenario and the setting with an odds ratio of 1.8, 44 noise SNPs, and a sample size of $N = 2000$.

Chapter 3

Efficient gene–environment interaction testing through bootstrap aggregating

In the following, the second manuscript [Lau et al., 2023], which was published in the journal Scientific Reports, is presented and addresses Research Gaps 1 and 3.

Efficient gene–environment interaction testing through bootstrap aggregating

Michael Lau, Sara Kress, Tamara Schikowski, and Holger Schwender

Authorship:	First author
Contributed part:	85%
Contribution:	Development of the statistical method Implementation in software Designing the simulation study Conceiving the real data analysis Conducting the statistical data analyses Preparing figures and tables Interpretation of results Writing the paper
Journal:	Scientific Reports
Impact factor:	4.562 (2022)
Date of publication:	17 January 2023
DOI:	10.1038/s41598-023-28172-4

scientific reports

OPEN

Efficient gene–environment interaction testing through bootstrap aggregating

Michael Lau[1,2]✉, Sara Kress[2], Tamara Schikowski[2] & Holger Schwender[1]

Gene–environment (GxE) interactions are an important and sophisticated component in the manifestation of complex phenotypes. Simple univariate tests lack statistical power due to the need for multiple testing adjustment and not incorporating potential interplay between several genetic loci. Approaches based on internally constructed genetic risk scores (GRS) require the partitioning of the available sample into training and testing data sets, thus, lowering the effective sample size for testing the GxE interaction itself. To overcome these issues, we propose a statistical test that employs bagging (bootstrap aggregating) in the GRS construction step and utilizes its out-of-bag prediction mechanism. This approach has the key advantage that the full available data set can be used for both constructing the GRS and testing the GxE interaction. To also incorporate interactions between genetic loci, we, furthermore, investigate if using random forests as the GRS construction method in GxE interaction testing further increases the statistical power. In a simulation study, we show that both novel procedures lead to a higher statistical power for detecting GxE interactions, while still controlling the type I error. The random-forests-based test outperforms a bagging-based test that uses the elastic net as its base learner in most scenarios. An application of the testing procedures to a real data set from a German cohort study suggests that there might be a GxE interaction involving exposure to air pollution regarding rheumatoid arthritis.

Many complex diseases are influenced by both genetic and environmental risk factors. Often, their effects are studied individually, e.g., in genome-wide association studies (GWAS) or environmental health studies. These kinds of analyses, thus, study main/marginal effects of the respective risk factor type, i.e., effects independent of the other risk factor type. However, it is well known that the genetic make-up and environmental risk factors can influence the risk of disease in an interplay[1]. This phenomenon is known as gene–environment (GxE) interaction and is present if, for different genotypes, different disease susceptibilities to an environmental factor are underlying. This is, for example, the case if an individual is particularly susceptible to certain environmental exposure if the individual carries a specific genetic variant. For example, if an individual has xeroderma pigmentosum—a genetic defect that decreases the ability to repair DNA damage caused by ultraviolet radiation—and is exposed to sunlight, the risk effect of developing skin cancer through sun radiation is magnified compared to individuals without this genetic defect[1].

Unveiling GxE interactions leads to a better understanding of the manifestation of complex diseases. Moreover, knowing specific GxE interactions could have a high impact on precision medicine by specifically protecting individuals that are highly susceptible to certain environmental health effects, i.e., performing individual risk prevention[2].

In practice, GxE interactions are tested using single SNPs (single nucleotide polymorphisms—counting the number of less frequent base-pair substitutions at a specific locus in the DNA) in linear or logistic regression models. However, testing single SNPs in parallel requires adjustment for multiple testing, which reduces the statistical power for detecting a GxE interaction. To avoid this problematic, a GRS-(genetic risk score)-based approach can also be employed for taking multiple loci at once into account[3]. GRS summarize genetic variants with respect to a specific phenotype to a single statistic. Their utility can, e.g., lie in uncovering biological relationships in the development of diseases or their utility can be clinical for individual risk prevention[4–6]. GRS can be constructed internally—by using the considered study sample also for constructing the GRS—or

[1]Mathematical Institute, Heinrich Heine University, Düsseldorf, Germany. [2]IUF – Leibniz Research Institute for Environmental Medicine, Düsseldorf, Germany. ✉email: michael.lau@hhu.de

externally—by using summary statistics of independent association studies. However, the external approach requires the availability of such summary statistics that match the considered outcome, the analyzed genomic region, and the considered population type, which might not be the case[3]. Moreover, by externally constructing GRS, only marginal genetic effects are considered, i.e., ignoring potential gene-gene interaction effects. Thus, we focus on the internal GRS construction approach in this article.

The GRS itself does not take any non-genetic variables into account. Thus, the variable which is used for interaction testing with the environmental term, the GRS, is a summary of genetic effects with respect to the outcome. However, for detecting GxE interactions, usually a GLM (generalized linear model) is fitted including potential confounders. For statistically testing the GxE interaction, typically a Wald test is employed.

The GRS approach leads to a major short coming. If an appropriate GRS model is not known beforehand, the available study data needs to be separated into two independent sub data sets, training data for constructing the GRS and test data for testing the GxE interaction. Therefore, the GxE interaction test cannot utilize the full available sample size, which reduces the statistical power for detecting a GxE interaction.

Several GxE interaction testing approaches have been proposed recently that avoid this data splitting problem[7,8]. Similar to the common GRS-based approach, SBERIA[9] (set-based gene–environment interaction test) constructs a weighted sum of SNPs for testing the GxE interaction. Another class of GxE interaction tests is given by variance component tests that test the variance of the true interaction coefficients. The interaction effects are identified with random effects such that testing whether the interaction effect coefficients are zero is equivalent to testing if the underlying effect variance is zero[7]. Established methods of this class include GESAT[10] (gene–environment set association test), iSKAT[11] (interaction sequence kernel association test), and MiSTi[12] (mixed effects score tests for interaction). Two-step GxE interaction testing procedures screen all considered SNPs and aggregate the positively screened SNPs to perform a global test for the presence of a GxE interaction among the considered SNPs in a second step[8]. The GxE interaction testing methods ADABF[13] (adaptive combination of Bayes factor method), EDGxE[14] (EG [environment-genotype] and DG [disease-genotype] screening with GxE interaction testing) and cocktail GxE interaction tests[15] are such two-step procedures.

In this article, we propose a GxE interaction testing approach that overcomes the data splitting problem while being able to model arbitrarily complex genetic effects and avoiding the multiple testing problem of the single-SNP-based test. Similar to the classical GRS-based test, our test also relies on modeling the genetic effect on the outcome through a GRS. Our approach can incorporate the full study data set for both training the GRS and testing the GxE interaction while still avoiding the overfitting problem. The idea consists of constructing the GRS via the ensemble method bagging (bootstrap aggregating)[16] and using its well-known OOB (out-of-bag) prediction mechanism for creating unbiased predictions on the whole training data.

Moreover, standard GRS construction methods such as the elastic net[3,17] can usually only incorporate marginal genetic effects and no gene-gene interaction effects. For example, it might be possible that the environment-response relationship is significantly altered only if specific genetic variants at two different loci are present at once, thus, leading to a GxE interaction involving a gene-gene interaction that cannot be covered by classical GRS approaches.

As prior analyses showed[6], using random forests instead of elastic net leads to a higher predictive ability of the GRS. Thus, we also propose using random forests[18] instead of elastic net as the GRS construction procedure in GxE interaction testing. This would yield a GxE interaction testing approach that is not restricted to simplifying assumptions on genetic effects and can incorporate every possible gene-gene interaction.

In this article, first, established GxE interaction testing procedures are discussed. Next, a novel testing approach based on bagging and OOB predictions is introduced. Moreover, an extension of this testing procedure using random forests is proposed. The methods are evaluated and compared to existing approaches in a simulation study that considers multiple realistic scenarios. As this analysis shows, the proposed testing procedures yield a higher statistical power than the reference testing procedures in many scenarios. Lastly, the methods are applied to a real epidemiological data set from a cohort study testing a GxE interaction involving air pollution regarding rheumatoid arthritis.

Methods

In the following, two standard approaches and three recently proposed approaches for testing GxE interactions are discussed first. Afterwards, we propose two novel approaches that can be used to overcome the drawbacks of the initially discussed methods.

Existing methods.

First, existing methods for GxE interaction testing are discussed.

Single-SNP-based GxE interaction test. The GxE interaction test based on single SNPs tests each considered SNP independently for a GxE interaction[3]. That is, for each $\mathrm{SNP_j}$, $j \in \{1, \ldots, p\}$, a GLM

$$g(\mathbb{E}[Y \mid \mathrm{SNP_j}, \mathrm{E}, \boldsymbol{C}]) = \beta_0 + \beta_1 \mathrm{SNP_j} + \beta_2 \mathrm{E} + \beta_3 \mathrm{SNP_j} \times \mathrm{E} + \sum_{\mathrm{i=1}}^{\mathrm{m}} \gamma_{\mathrm{i}} \mathrm{C_i} \tag{1}$$

is fitted, where also potential confounders $\boldsymbol{C} = \begin{pmatrix} C_1 & \ldots & C_m \end{pmatrix}$ are included in the model to adjust the main effects of $\mathrm{SNP_j}$ and the environmental variable E as well as their interaction effect for these variables. If a binary disease status is the considered outcome, logistic regression models via the link function $g = \mathrm{logit}$ are fitted. If the considered outcome is continuous, the identity link $g = \mathrm{Id}$ is used for fitting linear regression models.

In each of these models, the statistical hypothesis $H_0 : \beta_3 = 0$ versus $H_1 : \beta_3 \neq 0$ is tested, i.e., whether there is an interaction effect of the SNP and the environmental variable E on the outcome. A Wald test is usually performed for testing this hypothesis. Alternatively, a score test or a likelihood-ratio test can be carried out for testing the same hypothesis[19]. If, e.g., it should be tested whether a gene interacts with E, its SNPs are tested and the test decision for the whole gene is made by adjusting the individual SNP testing results for multiple testing. Usually, a Bonferroni correction is carried out[3,20]. If, after the Bonferroni correction, for at least one SNP the null hypothesis could be rejected, the global null hypothesis of no GxE interaction on the gene is rejected as well.

This approach has the advantage of not having to train a model but to directly perform statistical testing. Moreover, it is very simple, straightforward, and computationally feasible, since the individual tests can also be parallelized. However, due to considering individual SNPs and performing adjustment for multiple testing, statistical power for detecting a GxE interaction is lost.

GRS-based GxE interaction test. In contrast to the single SNP test, the GRS-based GxE interaction test aggregates multiple SNPs into one model and uses this model to test if there is an interaction in the considered genomic region[3]. Usually, the GRS is a linear combination of SNPs

$$\widehat{\mathrm{GRS}} = \hat{\alpha}_0 + \hat{\alpha}_1 \mathrm{SNP}_1 + \ldots + \hat{\alpha}_p \mathrm{SNP}_\mathrm{p} \tag{2}$$

that is either constructed internally or externally.

External GRS rely on summary statistics of independent studies and use the individual SNP effect sizes for determining their weights $\hat{\alpha}_1, \ldots, \hat{\alpha}_p$[21,22]. This approach, thus, requires the availability of appropriate study data, i.e., the same outcome, the same genomic region, and the same population type had to be analyzed[23]. Furthermore, the external approach only allows the construction of linear GRS, i.e., in general not taking interactions between genetic loci into account.

Alternatively, GRS can be constructed internally[3,24], which means that the available data has to be divided into independent training and test data sets. The GRS is constructed using the training data and evaluated on the test data. This data splitting is crucial to avoid overfitting, i.e., to avoid detecting effects that are solely made up of statistical noise and are recognized due to the model adapting to this statistical noise. The internal approach also allows to generalize the task of constructing GRS to a statistical learning problem, in which a function is to be fitted that maps the SNPs to the outcome and that does not necessarily have to be linear[6].

When internally constructing GRS, usually a GLM-based procedure such as the elastic net[25] is utilized[17,26]. The elastic net also fits a linear model (2)—yielding the weights $\hat{\alpha}_1, \ldots, \hat{\alpha}_p$ and intercept $\hat{\alpha}_0$—and regularizes the effect coefficients $\boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 & \ldots & \alpha_p \end{pmatrix}$. This is done by including the penalty term

$$R_\xi(\boldsymbol{\alpha}) = \frac{1}{2}(1-\xi)||\boldsymbol{\alpha}||_2^2 + \xi||\boldsymbol{\alpha}||_1$$

in the optimization problem

$$\min_{\alpha_0, \boldsymbol{\alpha}} \left\{ -\frac{1}{N}\ell(\alpha_0, \boldsymbol{\alpha}) + \lambda R_\xi(\boldsymbol{\alpha}) \right\},$$

in which ℓ is the log-likelihood function of the considered parameters, λ is the penalty strength, and $\xi \in [0,1]$ is a parameter controlling the balance between the L_1 penalty and the L_2 penalty, i.e., the lasso penalty[27] and the ridge penalty[28], respectively. The lasso penalty leads to shrinking the coefficients of unimportant SNPs to zero while the ridge penalty assigns similar weights to highly correlated SNPs, which, e.g., might be the case for SNPs in high LD (linkage disequilibrium). Thus, the elastic net simultaneously performs a variable selection and a properly handling of SNPs in high LD. However, as for GLMs, only marginal SNP effects are modeled if no prior knowledge about which loci might interact is available, which is usually the case.

After constructing the GRS on training data, predictions $\widehat{\mathrm{GRS}} = \hat{\alpha}_0 + \hat{\alpha}_1 \mathrm{SNP}_1 + \ldots + \mathrm{O}\!\!\!\alpha_\mathrm{p} \mathrm{SNP}_\mathrm{p}$ on independent test data are performed. These predicted values of the GRS for the subjects in the test data set are then used to fit the GLM

$$g(\mathbb{E}[Y \mid \widehat{\mathrm{GRS}}, E, \boldsymbol{C}]) = \beta_0 + \beta_1 \widehat{\mathrm{GRS}} + \beta_2 E + \beta_3 \widehat{\mathrm{GRS}} \times E + \sum_{i=1}^{m} \gamma_i C_i. \tag{3}$$

As for the single-SNP-based test, if a binary disease status is the phenotype of interest, the logit is used as link function g for fitting logistic regression models in both the GRS construction step and the GxE testing step. For continuous phenotypes, linear regression models are fitted using the identity as link function g.

For testing if the considered genomic region interacts with E regarding the outcome Y, the statistical test $H_0 : \beta_3 = 0$ versus $H_1 : \beta_3 \neq 0$ is performed. Similar to the single-SNP-based test, this hypothesis is most commonly tested using a Wald test. A GxE interaction is present if H_0 is rejected at a prespecified level of significance. In contrast to the single-SNP-based test, this test result directly reflects the desired test decision such that no adjustment for multiple testing has to be performed.

The drawback of this GRS-based testing approach is the requirement for splitting the available data into independent training and test data sets, where simulation studies suggest that a random 50:50 split should be used[23]. However, since in this case only 50% of the data can be used for actually testing the GxE interaction, substantial statistical power for detecting the GxE interaction is lost.

Set-based gene–environment interaction test. SBERIA[9] is a GxE interaction test that also utilizes a weighted sum of SNPs, similar to the GRS-based procedure. In SBERIA, all SNPs are univariately screened for either the association with the environmental factor or the association with the outcome. The results of this screening are used for constructing a weighted sum of SNPs. More precisely, this sum is constructed as $\widehat{\mathrm{GRS}} = w_1\mathrm{SNP}_1 + \ldots + \mathrm{w_p}\mathrm{SNP_p}$ with

$$w_j = \varepsilon + \begin{cases} 0, & \text{if } p \text{ value for } \mathrm{SNP_j} > \theta \\ -1, & \text{if } p \text{ value for } \mathrm{SNP_j} \leq \theta \text{ and correlation of } \mathrm{SNP_j} \text{ with E (or Y) negative} \\ +1, & \text{if } p \text{ value for } \mathrm{SNP_j} \leq \theta \text{ and correlation of } \mathrm{SNP_j} \text{ with E (or Y) positive.} \end{cases}$$

The offset ε is usually chosen as 0.0001 and the significance threshold θ as 0.1. The GxE interaction is then tested as in the GRS-based test using the GLM from Eq. (3). However, in contrast to the GRS-based test, the weighted sum utilized in SBERIA only considers the magnitude of the genetic effects to a limited extent. Nonetheless, through this limited modeling, the overfitting problem of the GRS-based testing does not arise such that the full data can be utilized for constructing the weighted sum and testing the GxE interaction even in low sample size scenarios.

Gene–environment set association test. GESAT[10] is a GxE interaction test that belongs to the class of variance component tests. In variance component tests, the GLM

$$g(\mathbb{E}[Y \mid \mathbf{SNP}, E, \boldsymbol{C}]) = \delta_0 + \boldsymbol{\delta}_1^T\mathbf{SNP} + \delta_2 E + \boldsymbol{\delta}_3^T\mathbf{SNP} \times E + \sum_{i=1}^{m} \gamma_i C_i$$

is considered for testing the GxE interaction, where $\mathbf{SNP} = \left(\mathrm{SNP}_1 \ \ldots \ \mathrm{SNP_p}\right)$ is the vector of all considered SNPs, $\boldsymbol{\delta}_1$ is the vector of corresponding main effects and $\boldsymbol{\delta}_3$ is the vector of corresponding GxE interaction effects. The GxE interaction effects are modeled as random effects with mean zero and a common variance $\tau \geq 0$. Testing the presence of a GxE interaction anywhere in the considered set of SNPs is now equivalent to testing $H_0 : \tau = 0$ versus $H_1 : \tau > 0$. In GESAT, a score test is used for testing this hypothesis. For computing the score test statistic, the main effects have to be estimated under the null model only incorporating main effects. In GESAT, this is done by applying ridge regression. The authors have shown that the score test statistic follows—under the null distribution of no GxE interaction—asymptotically a mixture of χ^2-distributions.

Adaptive combination of Bayes factor method. ADABF[13] is a recently proposed GxE interaction testing approach that tries to overcome the issues of classical tests, i.e., the need for data splitting or for too conservative multiple testing adjustment, by considering Bayes factors. ADABF starts by individually screening all considered SNPs for associations with the outcome. Only the $p_S \leq p$ SNPs passing this screening (e.g., only SNPs that are significantly associated with respect to a level of significance of 5%) are used for testing the GxE interaction itself. Similar to the single-SNP-based test, individual GLMs (see Eq. (1)) are fitted for each considered SNP. Then, Bayes factors

$$\mathrm{BF} = \frac{\mathbb{P}(\mathrm{data} \mid \mathrm{H}_1)}{\mathbb{P}(\mathrm{data} \mid \mathrm{H}_0)}$$

are computed for each SNP and the corresponding hypothesis $H_0 : \beta_3 = 0$ versus $H_1 : \beta_3 \neq 0$ of the GxE interaction coefficient of this SNP. Prior knowledge from previous studies is used for configuring the variance of the prior distributions of both the main effects and the GxE interaction effects. Since it is of interest to test the whole considered set of SNPs for a GxE interaction and not just single SNPs, the Bayes factors are combined into summary scores

$$S_k = \sum_{l=1}^{k} \log(\mathrm{BF}_{(l)})$$

with $\mathrm{BF}_{(l)}$ ($l \in \{1, \ldots, p_S\}$) being the decreasingly sorted Bayes factors for the considered SNPs such that $\mathrm{BF}_{(1)} \geq \ldots \geq \mathrm{BF}_{(\mathrm{p_S})}$. These summary scores are also computed under the null distribution of no GxE interaction, i.e., by randomly sampling GxE interaction effects from a multivariate normal distribution with mean zero (corresponding to no effect) and a covariance matrix incorporating LD (linkage disequilibrium) between the SNPs. Afterwards, the original summary scores and the sampled versions are compared for deriving p values for every $k \in \{1, \ldots, p_S\}$. Minima of these p values are computed for deriving a final p value that tests the global null hypothesis of no GxE interaction across all considered SNPs.

Bootstrap aggregating. To overcome the loss in statistical power through limited modeling or data splitting, we propose employing bagging (bootstrap aggregating)[16] for constructing the GRS in GxE interaction testing. Bagging is an ensemble approach that constructs B single models and combines them to one prediction model by averaging over the predictions of the individual models. The number of models B is chosen prior to fitting the model and should be set to a sufficiently high number such that more iterations do not considerably change the ensemble model. Each individual model is fitted by randomly drawing a bootstrap sample from the complete available data set, i.e., drawing N observations with replacement from a data set consisting of N observations, and using this sample to train the model such as a GLM via elastic net. A key property of bagging is that it reduces the variance of the predictions, thus, stabilizing the predictions[29].

Since in every iteration a bootstrap sample is used for training the model, there is complementary data left that was not used for training this sub model. These data are called OOB (out-of-bag) data. Utilizing this fact,

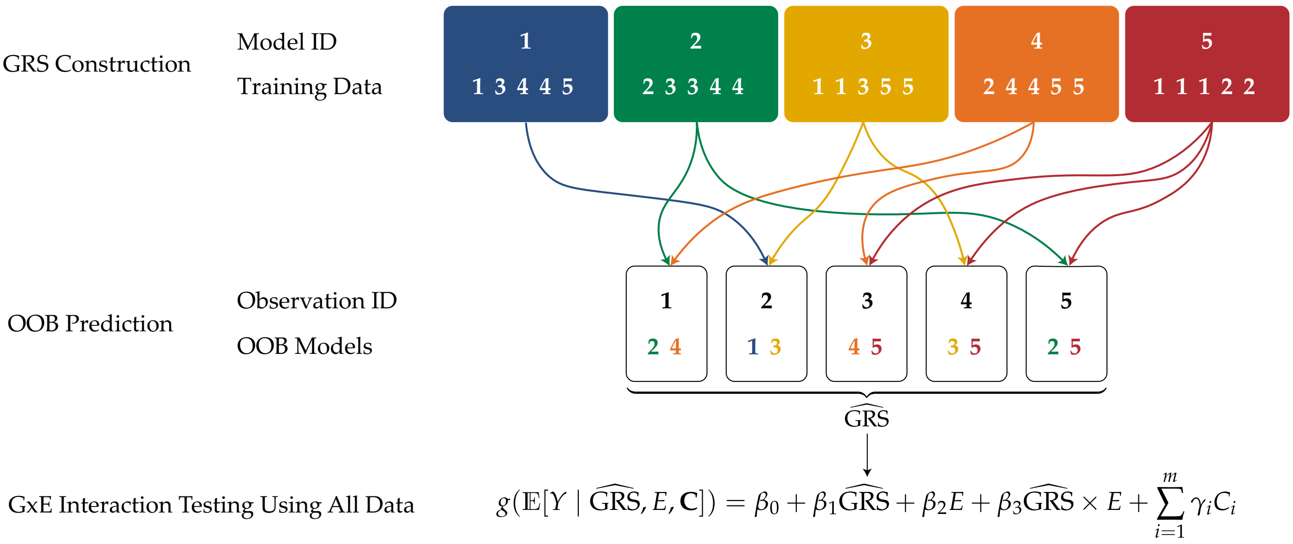
GRS Construction
Model ID
Training Data
1
1 3 4 4 5
2
2 3 3 4 4
3
1 1 3 5 5
4
2 4 4 5 5
5
1 1 1 2 2
OOB Prediction
Observation ID
OOB Models
1
2 4
2
1 3
3
4 5
4
3 5
5
2 5
$\widehat{\mathrm{GRS}}$
GxE Interaction Testing Using All Data
$g(\mathbb{E}[Y \mid \widehat{\mathrm{GRS}}, E, \mathbf{C}]) = \beta_0 + \beta_1\widehat{\mathrm{GRS}} + \beta_2 E + \beta_3\widehat{\mathrm{GRS}} \times E + \sum_{i=1}^{m} \gamma_i C_i$

Figure 1. Exemplary GxE interaction testing workflow utilizing bootstrap aggregating. $N = 5$ observations and $B = 5$ bagging iterations are considered.

unbiased predictions on the complete data set can be made. For each observation, those models are gathered that did not use this observation for training. These models are used to temporarily construct an ensemble and the OOB prediction is generated by calculating the average over these models, so that for an observation $(\boldsymbol{x}, y)$, its OOB prediction $\hat{y}_{\mathrm{OOB}}$ is calculated by

$$\hat{y}_{\mathrm{OOB}} = \frac{1}{|\mathcal{F}_{(\boldsymbol{x},y)}|} \sum_{f \in \mathcal{F}_{(\boldsymbol{x},y)}} f(\boldsymbol{x})$$

where

$$\mathcal{F}_{(\boldsymbol{x},y)} = \left\{f \in \mathcal{F} \mid (\boldsymbol{x}, y) \notin T_f\right\}$$

is the set of all trained models that did not use the considered observation for training, $\mathcal{F}$ is the set of all trained models in the ensemble, and T_f is the training data set used for training f. Thus, the OOB prediction for each observation is constructed by models that never have seen this specific observation, resembling test data predictions.

Proposed methods. In the following, two novel GxE interaction testing methods based on bagging are introduced.

GxE interaction testing through bagging. For avoiding the data splitting problem in GxE interaction testing, we propose constructing the GRS using bagging, e.g., bagging using the elastic net as the base learner, and computing the OOB prediction for all individuals in the whole data set. These predictions can then be used as a predictor in the GLM (3) as before and the statistical hypotheses $H_0 : \beta_3 = 0$ versus $H_1 : \beta_3 \neq 0$ are tested using a Wald test analogously to the conventional GRS-based test. Note that in contrast to the conventional GRS-based test, the GLM (3) is fitted and tested using all available data. Similarly, the GRS is in this case also fitted using all available observations. Therefore, neither the modeling step nor the testing step suffer from reduced sample sizes in this approach.

Figure 1 illustrates the proposed bagged GxE interaction testing approach considering $N = 5$ subjects and $B = 5$ bagging iterations/bootstrap samples (note that both numbers should actually be much higher; here, we only consider these small numbers for illustration purposes). First, for each out of $B = 5$ bagging iterations, a bootstrap sample is drawn from the original sample consisting of $N = 5$ observations and used for training the respective model such as a GLM through elastic net. For example, in the first iteration, the model is fitted using the observations 1, 3, 4, and 5. Next, for each of the $N = 5$ observations, those models are selected that did not use the respective observation for training and are used for generating the OOB predictions for the GRS. For example, for the first observation, the models 2 and 4 are used to predict its GRS by averaging their predictions, since observation 1 was not used for fitting the models 2 and 4. These predicted GRS values are then used as the values of the predictor to fit the GLM (3) and test whether the GxE interaction is associated with the outcome via its coefficient β_3 using a Wald test.

The GRS can be an arbitrary summary of genetic loci. Hence, if loci from multiple genes should be tested in a single GxE interaction test, the bagged GRS is constructed using all considered loci at once. For example in the real data application, a GxE interaction is tested for an association-based SNP selection that can potentially lead to loci from multiple different genes and for a gene-based SNP selection that was derived by considering multiple genes at once.

Random-forests-based GxE interaction test. Common GRS construction procedures such as the elastic net rely on linear modeling of genetic effects. Thus, these approaches usually model only marginal genetic effects unless prior knowledge about which loci might interact is available. Instead, more flexible modeling techniques such as random forests[18], which are theoretically able to model every possible interaction, can also be used to construct GRS. It has been previously shown[6] that these predictions can substantially outperform the standard method elastic net in the construction of GRS.

Therefore, we propose using random forests for the GRS construction step in testing GxE interactions. Random forests is an extension of bagging that uses decision trees[30] as its base learner. The individual decision trees are further randomized by selecting random subsets of the predictor set in the recursive fitting procedure. This additional randomization leads to an increased variance reduction. Due to employing bagging, random forests is a natural candidate for applying the OOB-predictions-based GxE interaction test discussed in the previous section. Here, the sub models that are used for computing OOB predictions are the individual randomized decision trees.

Ethics approval and consent to participate. The study was conducted in accordance to the declaration of Helsinki. The SALIA cohort study has been approved by the Ethics Committees of the Ruhr-University Bochum and the Heinrich Heine University Düsseldorf. Written informed consent was received from all participants.

Simulation study

For examining the proposed GxE interaction testing procedures based on bagging using elastic net and on random forests, respectively, we compared these procedures with each other, with the two classical testing approaches, i.e., the single-SNP-based test and the GRS-based test using elastic net, and with three recently proposed GxE interaction testing procedures, namely ADABF, GESAT, and SBERIA, in a simulation study considering several realistic data scenarios.

Simulation setup. In every simulation setting, 1000 independent replications were carried out, i.e., 1000 independent data sets were generated and evaluated for each considered study setting. The samples sizes were varied between $N = 500$, $N = 1000$, and $N = 2000$. 50 SNPs were simulated independently, resembling LD-based pruned SNPs, using random minor allele frequencies in the range of [0.15, 0.45], as in the analyses conducted by Lau et al.[6]. Similarly, dominant modes of inheritance were used for modeling the outcomes. The environmental term was generated by fitting a log-normal distribution on recorded exposures to nitrogen dioxide (NO_2) in the SALIA study[32] and randomly sampling from this fitted log-normal distribution. The SALIA study is described in more detail in the following section, in which the data from this study is also used in a real data application.

For the GRS-based testing approach employing elastic net, the data sets have to be divided into training and test data sets. Random 50:50 splits were used as recommended by Hüls et al.[23] in the context of GxE interaction testing. Additionally, a binary and a continuous outcome were simulated and analyzed. For the binary outcome, the prevalence, i.e., the probability of developing a disease without any exposure and genetic susceptibility, was chosen in each setting such that balanced data sets were generated, i.e., data sets, in which approximately half of the observations are cases and the other half are controls, which resembles (balanced) case-control studies.

The outcomes were generated following GLMs that are described in more detail below. Both the binary and the continuous outcomes used the same linear predictors. For the binary outcome, the inverse of the logit link function was used to generate case probabilities $\mathbb{P}(Y = 1 \mid \mathbf{SNP}, E)$ for randomly sampling the simulated outcome. For the continuous outcome, random noise from the standard normal distribution was added to each linear predictor $\mathbb{E}[Y \mid \mathbf{SNP}, E]$.

Type I error. First, the type I error rate of the testing procedures was evaluated, i.e., the probability of falsely rejecting the null hypothesis, which in this case is the probability of detecting a GxE interaction although no GxE interaction is present. In all cases, the typically used level of significance of 5% was considered. We, thus, checked if the proposed tests control the type I error rate at a level of 5%.

For evaluating the type I error rate, data sets were simulated by considering the model

$$g(\mathbb{E}[Y \mid \mathbf{SNP}, E]) = \alpha_0 + \alpha_1 \mathrm{SNP}_{1,\mathrm{D}} + \alpha_2 \mathrm{SNP}_{2,\mathrm{D}} + \alpha_3 \mathrm{SNP}_{3,\mathrm{D}} + \alpha_4 \mathrm{SNP}_{4,\mathrm{D}} + \alpha_{\mathrm{GxG}} \mathrm{SNP}_{1,\mathrm{D}} \mathrm{SNP}_{5,\mathrm{D}} + \alpha_{\mathrm{E}} \mathrm{E},$$

where $\mathrm{SNP}_{\mathrm{i,D}} := \mathbb{1}(\mathrm{SNP}_{\mathrm{i}} > 0)$ is a SNP exhibiting a dominant mode of inheritance. In this model, thus, no GxE interaction is present. The marginal genetic effect sizes $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \log(1.5)$ were fixed to an odds ratio of 1.5, resembling moderate effects. The gene-gene interaction effect was fixed to $\alpha_{\mathrm{GxG}} = 2\log(1.5) = \log(2.25)$. The marginal environmental effect was fixed to $\alpha_E = \log(1.02)$. The effect size for the environmental term may seem rather small compared to the genetic effect sizes. However, this is due to the fact that the environmental factor in the SALIA study attains higher values with a median of 23.91.

Power—different GxE interaction effect intensities and sample sizes. Next, we evaluated the statistical power of the proposed GxE interaction tests, where the power is the probability of correctly rejecting the null hypothesis, i.e., the probability of detecting a true GxE interaction.

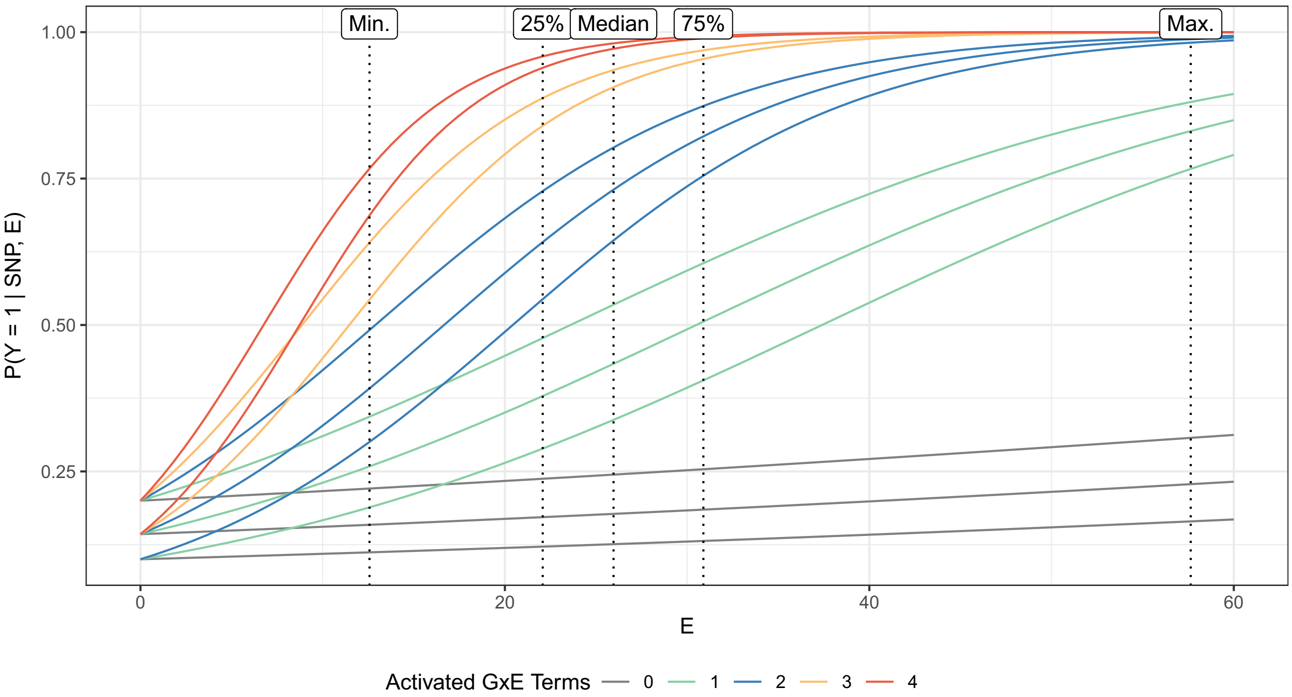

Figure 2. Design for simulating a binary phenotype. Exposure-response curves are depicted for different genotypes and a strong GxE interaction effect $\alpha_{\mathrm{GxE}} = \log(1.05)$ in Eq. (4). The colors illustrate the GxE interaction intensity referring to the number of activated GxE interaction terms in Eq. (4). The utilized distribution of the environmental factor is shown at the top by the 25%, 50%, and 75% quantiles and the minimum and the maximum.

For evaluating the power, the model

$$\begin{aligned} g(\mathbb{E}[Y \mid \mathbf{SNP}, E]) = {} & \alpha_0 + \alpha_1 \mathrm{SNP}_{1,\mathrm{D}} + \alpha_2 \mathrm{SNP}_{2,\mathrm{D}} + \alpha_{\mathrm{E}} \mathrm{E} \\ & + \alpha_{\mathrm{GxE}} \mathrm{SNP}_{3,\mathrm{D}} \mathrm{E} + \alpha_{\mathrm{GxE}} \mathrm{SNP}_{4,\mathrm{D}} \mathrm{E} + 2\alpha_{\mathrm{GxE}} \mathrm{SNP}_{1,\mathrm{D}} \mathrm{SNP}_{5,\mathrm{D}} \mathrm{E} \end{aligned} \tag{4}$$

was used for generating the data sets.

For choosing realistic parameters, we analyzed the true underlying models. The final parameter choice is illustrated in Fig. 2, which depicts the desired simulation setup through the true modeling probabilities $\mathbb{P}(Y = 1 \mid \mathbf{SNP}, E)$. The three curves at the bottom almost resembling a linear relationship correspond to the case of only the marginal environmental effect being active, i.e., no interacting risk allele being present that increases the slope. Thus, these curves share the same slope. The curves above are induced by interacting risk alleles being present and increasing the slope. In this model, almost the whole range in the probability spectrum is covered, i.e., probabilities $\mathbb{P}(Y = 1 \mid \mathbf{SNP}, E)$ of almost 100% if all risk alleles are present at once and a high exposure is given or probabilities $\mathbb{P}(Y = 1 \mid \mathbf{SNP}, E)$ of around 10% if no risk alleles and no exposure are given.

The corresponding model parameters are, therefore, chosen as follows. As in the type I error evaluation, the marginal genetic effects were fixed to $\alpha_1 = \alpha_2 = \log(1.5)$. The marginal environmental effect size was fixed to $\alpha_E = \log(1.01)$ and the effect size α_{GxE} of the GxE interaction involving SNP_3 and SNP_4 was varied between $\log(1.01)$, $\log(1.03)$, and $\log(1.05)$. The effect size $2\alpha_{\mathrm{GxE}}$ of the GxE interaction also incorporating a GxG interaction was doubled, since interaction effects are in general more difficult to capture, i.e., requiring more data or stronger effect sizes.

Power—main effects and different levels of statistical noise. For analyzing the methods' performances under the presence or absence of main effects and different levels of statistical noise, further simulations were conducted. The simulation setup considered by Lin et al.[13] was used as a basis for these simulations. Thus, the outcome was generated using the model

$$g(\mathbb{E}[Y \mid \mathbf{SNP}, E]) = \sum_{j=1}^{K} \alpha_j \mathrm{SNP}_{\mathrm{j,D}} + \sum_{\mathrm{j=0.5K+1}}^{1.5\mathrm{K}} \alpha_{\mathrm{GxE_j}} \mathrm{SNP}_{\mathrm{j,D}} \mathrm{E}. \tag{5}$$

Therefore, half of the interacting SNPs also exhibit main effects if the corresponding coefficients are unequal to zero. The number K of interacting SNPs and SNPs that may exhibit main effects was set to 10. The number of total SNPs was varied between 20, 50, and 100, simulating different settings of statistical noise by including more SNPs that have theoretically no effect on the outcome. The sample size was set to $N = 2000$, since different

Setting	Binary $\alpha_1, \ldots, \alpha_{10}$	Continuous $\alpha_1, \ldots, \alpha_{10}$	Binary $\alpha_{\mathrm{GxE}_6}, \ldots, \alpha_{\mathrm{GxE}_{15}}$	Continuous $\alpha_{\mathrm{GxE}_6}, \ldots, \alpha_{\mathrm{GxE}_{15}}$
1	0	0	$\pm[\log(1.2), \log(1.4)]$	$\pm[0.13, 0.17]$
2	0	0	$\pm[\log(1.4), \log(1.6)]$	$\pm[0.18, 0.22]$
3	$\pm[\log(1.2), \log(1.4)]$	$\pm[0.13, 0.17]$	$\pm[\log(1.2), \log(1.4)]$	$\pm[0.13, 0.17]$
4	$\pm[\log(1.4), \log(1.6)]$	$\pm[0.18, 0.22]$	$\pm[\log(1.4), \log(1.6)]$	$\pm[0.18, 0.22]$

Table 1. Simulation settings for the second simulation scenario (see Eq. (5)) considering different effect sizes, different levels of statistical noise, and the presence or absence of main effects.

samples sizes were already analyzed in the previously described simulation scenario. For every setting, 100 independent replications were conducted, i.e., 100 independent data sets were generated and evaluated for each considered setting in this additional simulation scenario. The SNPs were simulated analogously to the previous simulation scenario, i.e., independently (resembling LD-based pruned SNPs) and using random minor allele frequencies in the range of [0.15, 0.45]. Similar to Lin et al.[13], the environmental variable E was generated as a binary variable with $\mathbb{P}(E = 1) = \mathbb{P}(E = 0) = 0.5$.

Analogously to Lin et al.[13], the GxE interaction testing procedures were evaluated in four different simulation settings. These simulation settings are summarized in Table 1. First, two settings with no main effects, i.e., $\alpha_j = 0$ for all $j \in \{1, \ldots, K\}$, were evaluated. In these two settings, the effect sizes were varied. In the first setting, the coefficients were randomly drawn from the uniform distribution on $[\log(1.2), \log(1.4)]$ for the binary outcome and from the uniform distribution on [0.13, 0.17] for the continuous outcome. These effect sizes resemble small genetic effects. In the second setting, larger genetic effects were included. Hence, coefficients from the uniform distribution on $[\log(1.4), \log(1.6)]$ for the binary outcome and from the uniform distribution on [0.18, 0.22] for the continuous outcome were randomly drawn in the second simulation setting. Furthermore, two settings including main effects were evaluated. Here, the main effect and GxE interaction effects were randomly drawn according to the previously described uniform distributions. In all settings, the signs of the effect coefficients were randomly drawn. Thus, settings in which the main effect and the corresponding GxE interaction effect point in the same direction are covered as well as settings in which the main effect and the corresponding GxE interaction effect point in different directions are covered.

Application of the GxE interaction tests.

The application of the GRS-based GxE interaction testing procedures requires the choice of reasonable parameter settings for the underlying statistical learning method.

For fitting elastic net models, the strength of the penalty $\lambda \geq 0$ has to be tuned, which is usually done by k-fold cross-validation. In this article, 10-fold cross-validation was employed throughout all analyses. The balance parameter ξ was fixed to 0.5, as 0.5 is a reasonable value in most situations for constructing GRS[3]. The R[31] software package `glmnet`[33] was used for fitting elastic net models.

For the novel bagging-based GxE interaction tests, the number B of bagging iterations has to be chosen. In this article, we set $B = 500$, since this is a relatively high number of bagging iterations, such that more iterations would not considerably alter the ensemble.

For fitting random forests, the `R` software package `ranger`[34] was used. For the number of random variables drawn for evaluating tree splits, the standard setting of random forests for a higher number of predictors was used, i.e., `mtry` was set to $\lfloor p/3 \rfloor$, where p is the number of SNPs. The minimum number of observations contained in a terminal node (`min.node.size`) was set to $\lfloor 0.05 \times n_{\text{tree}} \rfloor$, in which n_{tree} is the number of observations one single tree uses for training. If bootstrap sampling is performed, $n_{\text{tree}} = N$, in which N is the total sample size, holds. This setting was used to avoid too deep trees that overfit and to fit trees that hold stable risk estimates in their leaves, as suggested by Malley et al.[35]. The number of trees in a random forest (`num.trees`) was set to 500, the standard setting in `ranger`.

ADABF tests were carried out using the standard settings and the corresponding code that is available online (https://homepage.ntu.edu.tw/~linwy/ADABFGE.html). GESAT tests were conducted utilizing the `R` package `iSKAT` (https://github.com/lin-lab/iSKAT-GESAT) using its standard settings. Due to lack of publicly available software, the SBERIA test was implemented manually and carried out using 0.0001 as the intercept for non-significant SNPs and 0.1 as the p value threshold, as proposed by Jiao et al.[9].

Results of the simulation study.

In the following, the results of the simulation study are presented.

Type I error. Figure 3 shows the estimated type I error rates for the considered methodologies. The red dashed line indicates the targeted 5% level. Both the bagged test using elastic net and random-forests-based test induce type I error rates that are around this level for both binary and continuous outcomes and smaller to larger data sets. Thus, the proposed methods seem to control the type I error. Similarly, the reference testing procedures based on single SNPs and elastic net regression also yield type I error rates around the 5% level. The alternative GxE interaction testing approaches ADABF and SBERIA induce type I error rates around 5% as well. However, in our simulation study, GESAT yields a type I error rate of over 10% for small samples and a binary outcome. Also for larger sample sizes and a binary outcome, GESAT induces higher type I error rates than the other methods. This issue might be caused by asymptotics that have not been reached due to the small sample size but

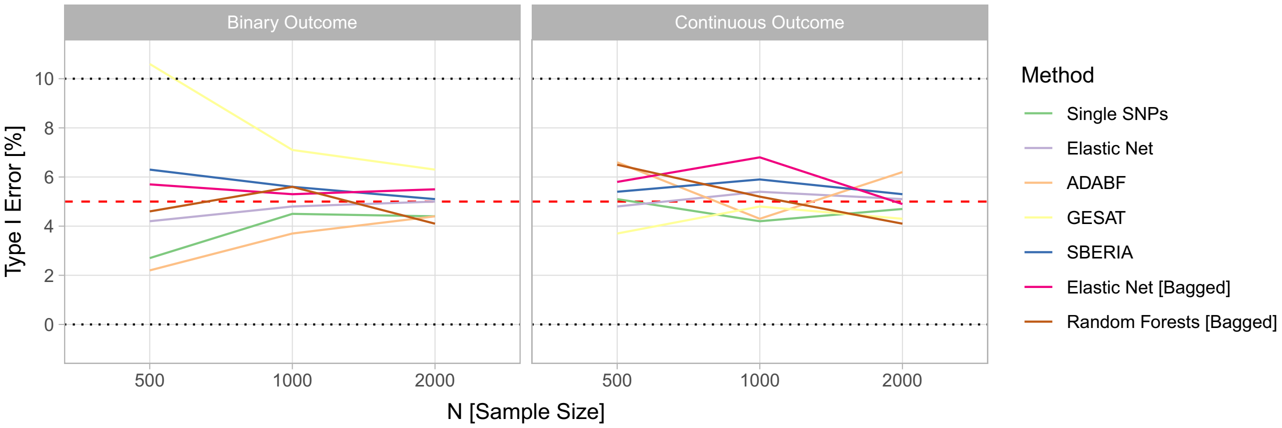

Figure 3. Type I error rates of the single SNP test, the GRS-based test using elastic net, ADABF, GESAT, SBERIA, the bagged GRS-based test using elastic net, and the random-forests-based test for testing GxE interactions in the simulation study.

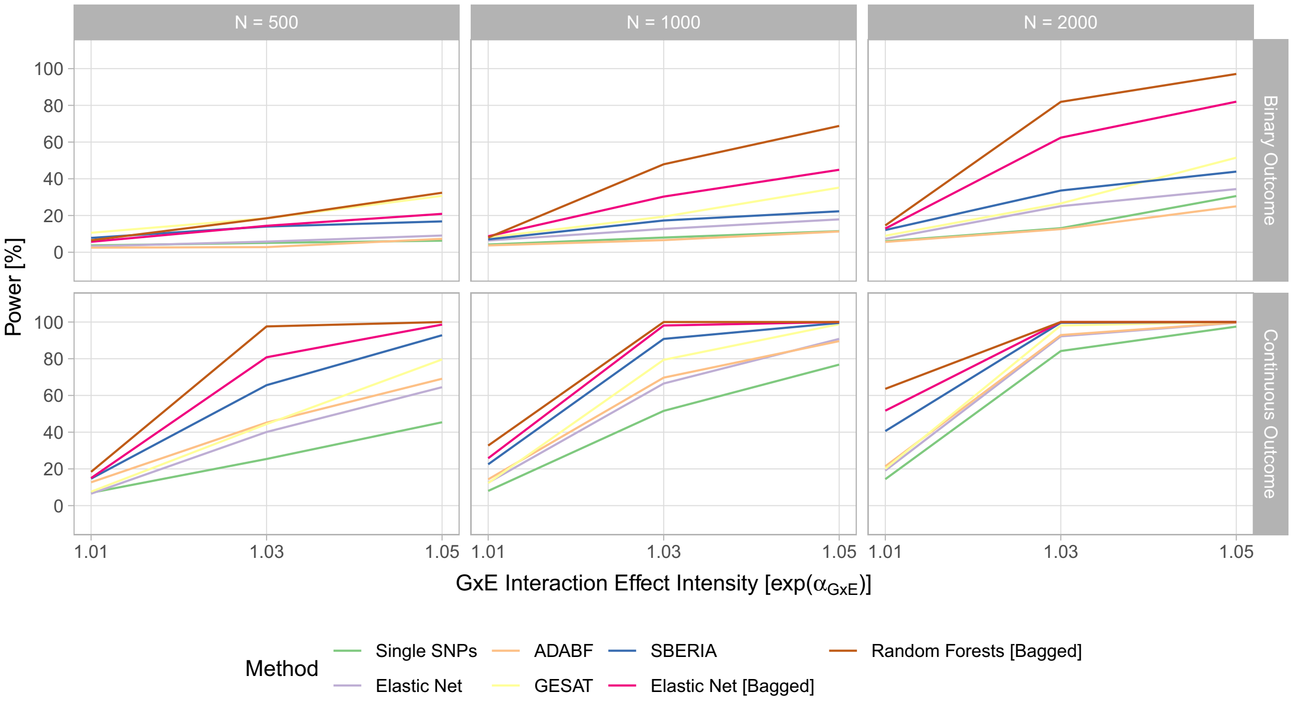

Figure 4. Statistical power of the single SNP test, the GRS-based test using elastic net, ADABF, GESAT, SBERIA, the bagged GRS-based test using elastic net, and the random-forests-based test for testing GxE interactions in the first scenario (see Eq. (4)) of the simulation study.

are vital to the theory of the GESAT test, in particular, for the distribution under the null hypothesis of no GxE interaction.

Power—different GxE interaction effect intensities and sample sizes. In Fig. 4, the results for the power evaluation of the first simulation scenario considering different GxE interaction effect intensities, different sample sizes, and a continuous environmental factor are shown. Unsurprisingly, the power rises with the available sample size and with the GxE effect intensity for all considered methods. For a large sample size and a strong GxE interaction effect, a power of 100% is reached, while for a small sample size and a weak GxE interaction effect, the power is around the prespecified tolerated type I error level. Therefore, the simulation design covers also scenarios in which the GxE interaction effect is almost undetectable and scenarios in which the GxE interaction should be detected, which was desired.

Regarding the comparison between the individual testing approaches, the single-SNP-based test seems to yield the lowest statistical power in most settings. For a continuous outcome, the GRS-based test, GESAT, and ADABF induce similar results. For a binary outcome, GESAT induces the highest power among these three tests. SBERIA yields the highest statistical power among the considered reference GxE interaction testing procedures.

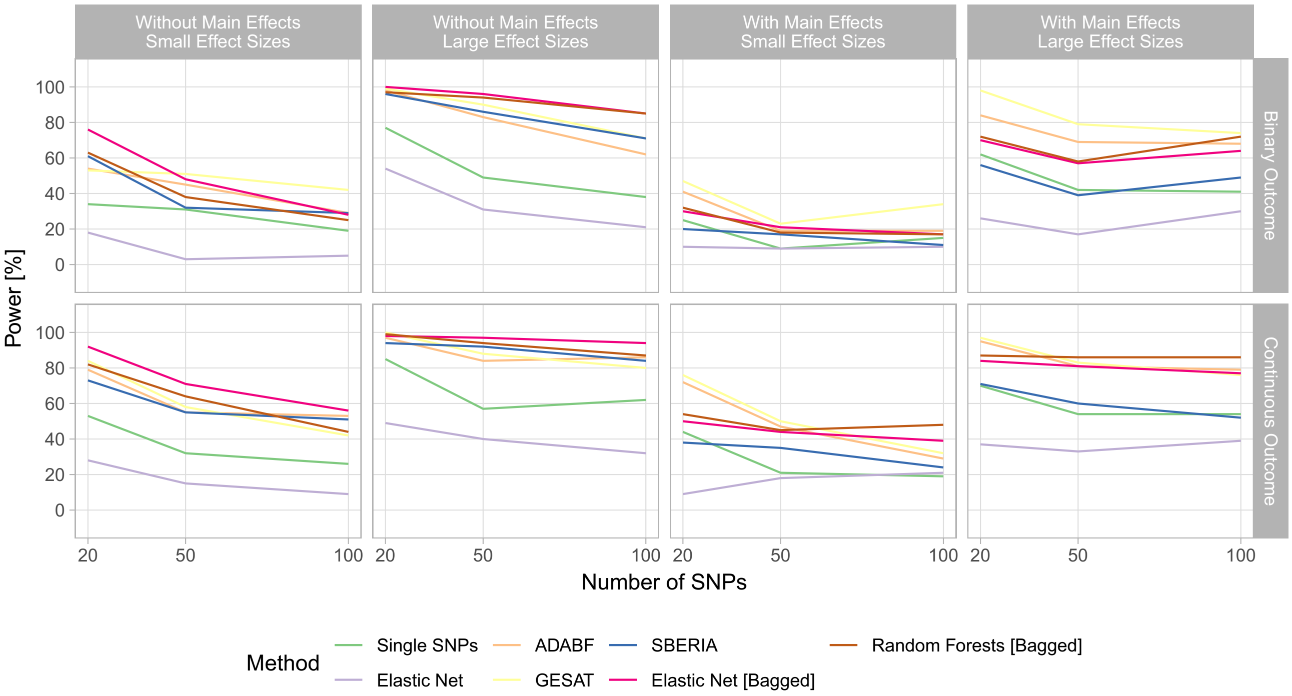

Figure 5. Statistical power of the single SNP test, the GRS-based test using elastic net, ADABF, GESAT, SBERIA, the bagged GRS-based test using elastic net, and the random-forests-based test for testing GxE interactions in the second scenario (see Eq. (5)) of the simulation study.

The newly proposed proposed procedure based on bagging and using elastic net as its base learner consistently achieves superior results to the reference approaches. The novel random-forests-based method induces an even higher statistical power than the bagging-based test that uses elastic net.

Thus, regardless of considering a binary or a continuous outcome, small or large samples, or weak or strong GxE interaction effects, the two tests based on bagging induce a comparatively high statistical power.

Power—main effects and different levels of statistical noise. Considering the second simulation scenario analyzing different levels of statistical noise, the presence or absence of main effects, differing effect sizes, and a dichotomous environmental risk factor, Fig. 5 depicts the statistical power achieved by the considered GxE interaction testing methodologies. The GRS-based GxE interaction test employing elastic net and the single-SNP-based test induce the lowest statistical power in most settings. When no main effects are present, SBERIA seems to yield better results than the two classical testing approaches. GESAT and ADABF induce a similar statistical power that is higher than the statistical power induced by the other methods—including the proposed methods—when considering settings with main effects and a low number of additional noise SNPs. The proposed bagging-based testing approaches yield a comparatively high statistical power in all settings. When considering settings without main effects or settings with main effects and a higher number of noise SNPs, the power of the bagging-based tests is particularly high.

The bagging-based test employing elastic net as its base learner seems to induce a slightly higher power than the random-forests-based test when considering settings without main effects. Vice versa, with main effects, random forests yields a slightly higher power. In general, the induced statistical power by the bagging-based tests is similar in this simulation scenario. The bagging-based test with elastic net presumably yields a higher power in this scenario (see Eq. (5)) due to considering a pure linear relationship compared to the previous scenario (see Eq. (4)), in which a gene-gene interaction was also present.

The proposed bagging-based tests seem to be more robust against a higher number of noise SNPs, since their statistical power does not severely decrease in comparison to the other procedures. Only for the first setting without main effects and small effect sizes, their statistical power considerably decreases for a higher number of noise SNPs.

Real data application

To verify the results of the simulation study and the applicability of the two proposed GxE interaction tests, a real data set from a German cohort study, the SALIA study (**S**tudy on the Influence of **A**ir Pollution on **L**ung, **I**nflammation and **A**ging)[32], was used for investigating GxE interactions.

At baseline, the SALIA study was conducted between 1985–1994 and included 4874 women aged between 54 and 55 years at their baseline examination. The study region included highly and less industrialized areas in North Rhine-Westphalia, Germany. In 2006, 4027 study participants completed a follow-up questionnaire about the diagnosis of chronic diseases. A further follow-up involving clinical examinations was conducted in

2007–2010, where genetic data was collected. Genome-wide genotyping was performed using the Axiom Precision Medicine Research Array GRCh37/hg19 (Affymetrix, Santa Clara, CA, USA). Imputation of unobserved genotypes using the Haplotype Reference Consortium[36] as reference panel on the Michigan Imputation Server[37] and quality controls[38] were performed. Individual exposures to air pollutants such as NO_2 during the first follow-up examinations were assessed using land-use regression models as part of the ESCAPE (European Study of Cohorts for Air Pollution Effects) project[39,40].

In the questionnaire of the first follow-up examination, the study participants were asked if they had a diagnosed rheumatic disease. Therefore, prevalent rheumatic diseases were considered as outcome in this article. Among the 560 women, 144 women stated they had a diagnosed rheumatic disease so that 416 women stated they did not have a rheumatic disease. Since rheumatoid arthritis is the most common rheumatic disease besides osteoarthritis[41–43], we focused on rheumatoid arthritis.

The data set analyzed in this article was restricted to subjects with available genotype data and information on the presence of rheumatic diseases. Thus, the analyzed data set consists of data from 560 women.

Gene ATLAS[44] was used for selecting SNPs that are significantly associated with the development of rheumatoid arthritis in the UK Biobank[45] (data field 20002). In particular, all SNPs that reached a level of significance of 10^{-80} were selected, which resulted in 91 SNPs in total. Canela-Xandri et al.[44] computed these p values by performing two-sided t-tests for each SNP on the residuals of linear mixed-effects models that were fitted for each trait and include potential confounders such as sex or age as fixed effects and a random effect adjusting for the population structure. The significance threshold was chosen such that about 100 SNPs were selected. 87 of these 91 SNPs were available in the analyzed data set from the SALIA cohort study. A detailed list of the analyzed SNPs can be found in Supplementary Table S1. This first SNP selection is based on single SNPs that showed a significant association with the disease phenotype of rheumatoid arthritis.

Moreover, we also considered a gene-based SNP selection for confirming the applicability of the proposed GxE interaction tests in gene-based analyses. Analogously to Lau et al.[6], the three genes HLA-DRB1, HLA-DPB1, and HLA-DOA from the human leukocyte antigen (HLA) class II complex were chosen, since they seem to explain a large fraction of the heritability of rheumatoid arthritis in the HLA class II complex[46–51]. All available SNPs from these three genes were selected, which resulted first in 385 SNPs. These SNPs were then clumped based on LD (linkage disequilibrium)[52] considering $r^2 = 0.5$ using PLINK version 1.9[53]. The LD-based clumping resulted in 72 tag SNPs. This set of 72 gene-based selected SNPs and the set of 87 association-based selected SNPs are disjoint such that there is no single SNP that is present in both sets.

It has already been shown that an interaction between genetic risk factors and smoking exists in the development of rheumatoid arthritis[54,55]. Thus, it can be suspected that traffic-related air pollution such as NO_2 might also be involved in a GxE interaction, which is analyzed in the following. Hence, for testing the presence of a GxE interaction, we considered the exposure to NO_2 as the environmental variable potentially interacting with genetic risk factors.

Adjustment for relevant potential confounders was performed using the same set of potential confounders as Hüls et al.[56] in their GxE interaction analysis. In particular, the genetic and environmental marginal effects as well as the GxE interaction effect were adjusted for subject age, socioeconomic status, BMI, smoking status, passive smoking, and household heating by indoor combustion of fossil fuels.

For more details about the SALIA study itself and an analysis of rheumatic diseases in the SALIA study, see Krämer et al.[57] and Lau et al.[6], respectively.

The evaluated GxE interaction methods were applied analogously to the simulation study. For application details, see Section "Application of the GxE interaction tests".

Results of the real data application.

Figure 6 summarizes the results of the real data analysis considering the association-based SNP selection by the induced p values of the considered methodologies. The single-SNP-based test yields a p value of 1 due to the Bonferroni correction, i.e., none out of the 87 SNPs yields a significant GxE interaction. Without the Bonferroni correction, the single-SNP-based test would yield a p value of 0.125, which is still not significant with respect to a level of significance of 5%. However, note that not correcting for multiple testing would inflate the type I error rate, which would disqualify the single-SNP-based test as a valid statistical test.

The common GRS-based test employing elastic net yields a median p value of about 0.8 such that, in almost all repetitions, no GxE interaction was detected. However, the resulting p value heavily varies between the replications. This variance is induced by random data splits for training the GRS and testing the GxE interaction and by cross validation that randomly splits the respective training data set for choosing the ideal elastic net regularization penalty.

ADABF yields a median p value of 0.47—not indicating a GxE interaction.

GESAT and SBERIA induce p values of 0.19 and 0.24, respectively, which are substantially lower than the p values of the other reference testing procedures. However, the null hypothesis of no GxE interaction cannot be rejected with respect to a level of significance of 5%. Note that GESAT was not applied to the original set of 87 SNPs but to a LD-based pruned (using $r^2 = 0.975$) set consisting of 37 SNPs, since the GESAT software could not be applied to the original SNP selection, which seemed to be due to the very high LD of some of the SNPs.

The novel bagging-based test utilizing elastic net yields a median p value of about 0.12, which is not significant with respect to a level of significance of 5%, however, considerably lower than the median p value induced by the common tests. In no iteration, this test could detect a GxE interaction. Nonetheless, the variance of the resulting p values seems to be almost completely diminished in contrast to the common elastic-net-based test that does not employ bagging.

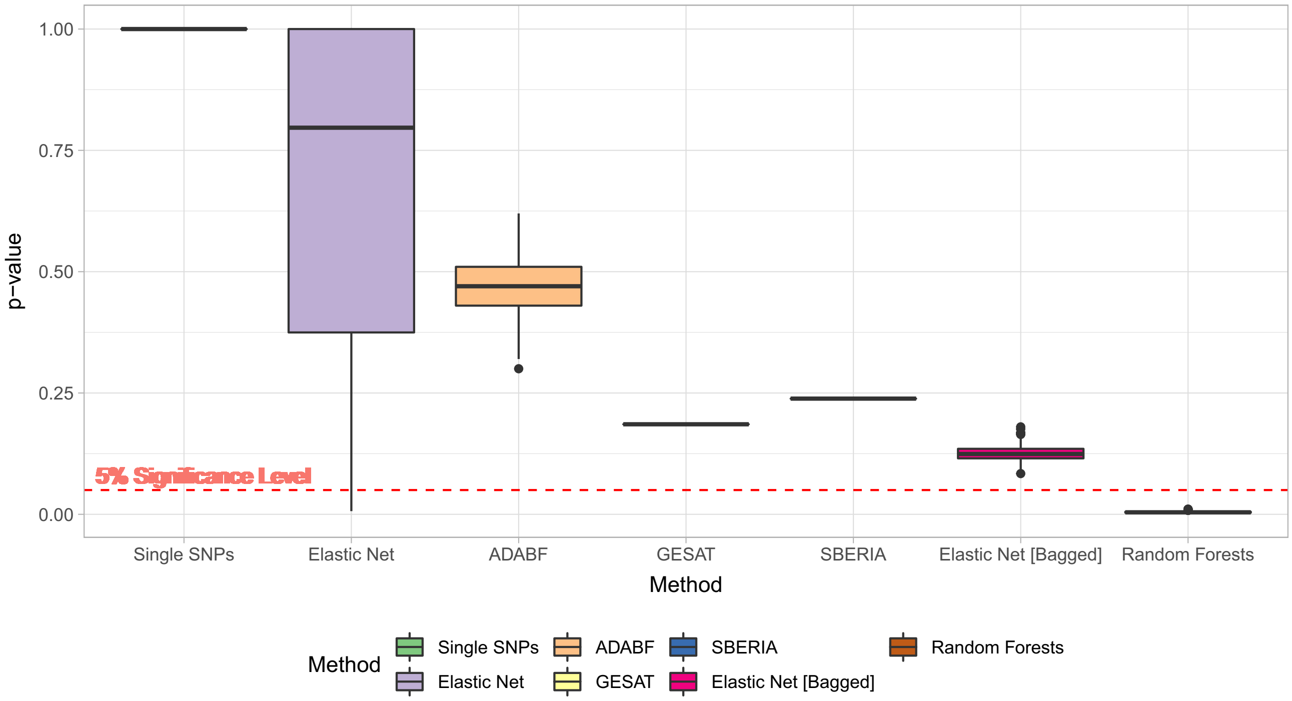

Figure 6. *p* values of 1000 independent applications of the GxE interaction testing procedures to the considered real data set from the SALIA cohort study analyzing the association-based SNP selection containing 87 SNPs. For elastic net, the train/test data splits changed. For the two bagging-based tests, the bootstrap samples changed. For ADABF, the random sampling from the null distribution of GxE interaction coefficients changed. The single-SNP-based test, GESAT, and SBERIA were applied only once, since there is no randomness involved in the application of these tests.

Using random forests, a median *p* value of about 0.004 is yielded, which is by far the lowest. In every repetition, the random-forests-based test rejects the null hypothesis of no GxE interaction. Thus, this suggests that there might be a GxE interaction between genetic risk factors and air pollution exposure regarding rheumatoid arthritis.

The results of the additional gene-based analysis are depicted in Supplementary Fig. S1. For these genes, none of the GxE interaction tests indicates the presence of a GxE interaction. In this analysis, random forests yields similar *p* values to ADABF and SBERIA, while the bagging-based test with elastic net as its base learner yields similar *p* values to GESAT and the single-SNP-based test.

Discussion

In this article, we proposed a novel GxE interaction testing approach utilizing bagging and its OOB prediction mechanism. We further proposed using random forests as the GRS construction method in GxE interaction testing. The main advantage of these novel tests is that they allow to utilize all subjects in both the GRS construction and the GxE interaction testing in contrast to classical procedures. Furthermore, this general approach allows utilizing statistical learning procedures that can model more complex patterns such as decision trees in random forests.

The new methods were first compared to two commonly used procedures, the single-SNP-based test and standard GRS-based test, and three recently proposed procedures, ADABF, GESAT, and SBERIA, in a simulation study considering both binary and continuous outcomes as well as different sample sizes, GxE interaction effect sizes, different levels of statistical noise, and the presence or absence of main effects. The analyses were started by evaluating the type I error rate, i.e., the probability of detecting a false GxE interaction, to see whether the proposed methods are valid statistical tests. Both tests could control the type I error with respect to the prespecified significance level. The analyses were continued by evaluating the statistical power, i.e., the probability of detecting a true GxE interaction. Here, it could be observed that the proposed methods could induce strong results compared to the reference tests in most scenarios. In particular, for small sample sizes, in presence of gene-gene interactions, for high intensities of statistical noise, or in absence of main effects, the proposed tests induced a superior statistical power compared to the other considered tests. The random-forests-based test also yielded a considerably higher statistical power than the bagging-based test using elastic net as its base learner in most settings. In a real data application, a GxE interaction regarding rheumatoid arthritis involving the exposure to NO_2 was analyzed. The two novel methods induced the lowest *p* values. The random-forests-based test was the only test to consistently induce *p* values below the prespecified significance threshold, which suggests that there might be a GxE interaction.

The strength of the proposed tests can be largely explained due to the increased sample size for both constructing the GRS and testing the GxE interaction compared to the standard GRS-based test. However, the variance reducing property of bagging presumably also lead to improved GRS models that had a stronger association with the analyzed phenotype, which also likely increased the statistical power. A variance reduction could be seen in the real data application, where the bagging-based test using elastic net as its base learner considerably reduced the p value variance compared to the common elastic-net-based test without bagging. Here, the variance was reduced due to no longer requiring random train-test data splits and through bagging, reducing the variance induced by cross validation in fitting elastic net models. Moreover, the random-forests-based test's superior performance was also caused by the ability to detect gene-gene interactions, which is usually not possible with elastic net, and the increased variance reduction due to further randomizing the model fitting procedure. In the second simulation scenario considering no gene-gene interactions, performances of both bagging-based tests were similar. Therefore, there seems to be no drawback of using the random-forests-based test over the bagging-based test using elastic net as its base learner. These findings are in line with the analyses by Lau et al.[6], which showed that the predictive performance of a GRS constructed by random forests could compete with the predictive performance of a GRS constructed by elastic net, even when considering no gene-gene interaction effects.

The recently proposed GxE interaction testing approaches ADABF, SBERIA, and GESAT also do not rely on data splitting that is required by the conventional GRS-based GxE interaction test. However, our proposed bagging-based test offers the advantage of being able to capture arbitrarily complex genetic effects through the statistical learning procedure that is employed as base learner. In contrast to ADABF and SBERIA, our proposed methods do not explicitly perform variable selection before testing the GxE interaction. It might be that certain SNPs are excluded in the individual bagging iterations. However, these selections might only be valid for individual iterations and not for the ensemble model such that most, if not all, variables are most likely included in the GRS for testing the GxE interaction in some way. Nonetheless, due to the explicit regularization performed by elastic net or the implicit regularization performed by random forests through randomization[58], possibly uninformative SNPs should not considerably decrease the statistical power, as could be seen in the simulation study.

As discussed by Janitza and Hornung[59] and Mitchell[60], the OOB error in random forests can be biased in the sense that it overestimates the actual test error. To eliminate this bias, they suggest to perform subsampling without replacement instead of bootstrap sampling. In this case, the number of observations in a subsample drawn is set to about $0.632 \times N$, the asymptotic number of unique observations drawn when performing bootstrapping. For evaluating if sampling without replacement would further improve the performance of our proposed GxE interaction testing procedures, we repeated the analyses with sampling without replacement. The results are shown in Supplementary Fig. S2–S4 and are in line with the evaluations using sampling with replacement. Hence, no considerable difference could be observed.

With GxE interaction tests that perform a SNP selection prior to testing the GxE interaction itself such as the single-SNP-based test, the GRS-based test employing elastic net or the lasso, ADABF, or SBERIA, it is relatively simple to deduce which SNPs among all initially considered SNPs are likely to be responsible for a detected GxE interaction. With the bagging-based approach, the GRS becomes an ensemble of many models such that it is not obvious how to infer the subset of SNPs responsible for a detected GxE interaction. Nonetheless, in future research, the proposed methodology could be extended to be able to score which genetic loci influence the constructed model the most, e.g., by employing VIMs (variable importance measures).

In our evaluations, we used fixed hyperparameter settings for fitting random forests. However, especially the parameter for determining the number of random variables selected as potential splitting variables and the parameter for bounding the minimum number of observations contained in a single leaf can have a substantial impact on the performance of random forests. Thus, the statistical power of the random-forests-based test could potentially be further enhanced by conducting proper hyperparameter tuning.

Conclusion

As the simulation study showed, both proposed bagging-based testing procedures control the type I error, making them valid statistical testing procedures. Moreover, the bagging-based procedures induce a high statistical power for detecting GxE interactions compared to established GxE interaction tests. The novel random-forests-based test was the best GxE interaction testing method among all evaluated tests in many scenarios. In the real data application, the random-forests-based test detected a statistically significant GxE interaction regarding rheumatoid arthritis using NO_2 exposure as the environmental variable.

Data availability

The simulated data sets analyzed in this article are available from the corresponding author on reasonable request.

Code availability

The proposed methods are implemented and publicly available in the R[31] package GRSxE on CRAN.

Received: 22 November 2022; Accepted: 13 January 2023
Published online: 17 January 2023

References

1. Ottman, R. Gene–environment interaction: Definitions and study design. *Prev. Med.* **25**, 764–770. https://doi.org/10.1006/pmed.1996.0117 (1996).
2. Nakamura, S. *et al.* Gene–environment interactions in obesity: Implication for future applications in preventive medicine. *J. Hum. Genet.* **61**, 317–322. https://doi.org/10.1038/jhg.2015.148 (2016).

3. Hüls, A., Ickstadt, K., Schikowski, T. & Krämer, U. Detection of gene–environment interactions in the presence of linkage disequilibrium and noise by using genetic risk scores with internal weights from elastic net regression. *BMC Genet.* **18**, 55. https://doi.org/10.1186/s12863-017-0519-1 (2017).
4. Torkamani, A., Wineinger, N. E. & Topol, E. J. The personal and clinical utility of polygenic risk scores. *Nat. Rev. Genet.* **19**, 581–590. https://doi.org/10.1038/s41576-018-0018-x (2018).
5. Wray, N. R. *et al.* From basic science to clinical application of polygenic risk scores: A primer. *JAMA Psychiat.* **78**, 101–109. https://doi.org/10.1001/jamapsychiatry.2020.3049 (2021).
6. Lau, M., Wigmann, C., Kress, S., Schikowski, T. & Schwender, H. Evaluation of tree-based statistical learning methods for constructing genetic risk scores. *BMC Bioinformatics* **23**, 97. https://doi.org/10.1186/s12859-022-04634-w (2022).
7. Lin, W.-Y., Huang, C.-C., Liu, Y.-L., Tsai, S.-J. & Kuo, P.-H. Genome-wide gene–environment interaction analysis using set-based association tests. *Front. Genet.* **9**, 715. https://doi.org/10.3389/fgene.2018.00715 (2019).
8. Gauderman, W. J. *et al.* Update on the state of the science for analytical methods for gene–environment interactions. *Am. J. Epidemiol.* **186**, 762–770. https://doi.org/10.1093/aje/kwx228 (2017).
9. Jiao, S. *et al.* SBERIA: Set-based gene–environment interaction test for rare and common variants in complex diseases. *Genet. Epidemiol.* **37**, 452–464. https://doi.org/10.1002/gepi.21735 (2013).
10. Lin, X., Lee, S., Christiani, D. C. & Lin, X. Test for interactions between a genetic marker set and environment in generalized linear models. *Biostatistics* **14**, 667–681. https://doi.org/10.1093/biostatistics/kxt006 (2013).
11. Lin, X. *et al.* Test for rare variants by environment interactions in sequencing association studies. *Biometrics* **72**, 156–164. https://doi.org/10.1111/biom.12368 (2016).
12. Su, Y.-R., Di, C.-Z., Hsu, L., Genetics and Epidemiology of Colorectal Cancer Consortium. A unified powerful set-based test for sequencing data analysis of GxE interactions. *Biostatistics* **18**, 119–131. https://doi.org/10.1093/biostatistics/kxw034 (2016).
13. Lin, W.-Y., Huang, C.-C., Liu, Y.-L., Tsai, S.-J. & Kuo, P.-H. Polygenic approaches to detect gene–environment interactions when external information is unavailable. *Brief. Bioinform.* **20**, 2236–2252. https://doi.org/10.1093/bib/bby086 (2019).
14. Gauderman, W. J., Zhang, P., Morrison, J. L. & Lewinger, J. P. Finding novel genes by testing G × E interactions in a genome-wide association study. *Genet. Epidemiol.* **37**, 603–613. https://doi.org/10.1002/gepi.21748 (2013).
15. Hsu, L. *et al.* Powerful cocktail methods for detecting genome-wide gene–environment interaction. *Genet. Epidemiol.* **36**, 183–194. https://doi.org/10.1002/gepi.21610 (2012).
16. Breiman, L. Bagging predictors. *Mach. Learn.* **24**, 123–140. https://doi.org/10.1007/BF00058655 (1996).
17. Privé, F., Aschard, H. & Blum, M. G. B. Efficient implementation of penalized regression for genetic risk prediction. *Genetics* **212**, 65–74. https://doi.org/10.1534/genetics.119.302019 (2019).
18. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32. https://doi.org/10.1023/A:1010933404324 (2001).
19. Agresti, A. *Foundations of Linear and Generalized Linear Models* (Wiley, Hoboken, 2015).
20. Majumdar, A. *et al.* A two-step approach to testing overall effect of gene–environment interaction for multiple phenotypes. *Bioinformatics* **36**, 5640–5648. https://doi.org/10.1093/bioinformatics/btaa1083 (2021).
21. Choi, S. W., Mak, T.S.-H. & O'Reilly, P. F. Tutorial: A guide to performing polygenic risk score analyses. *Nat. Protoc.* **15**, 2759–2772. https://doi.org/10.1038/s41596-020-0353-1 (2020).
22. Che, R. & Motsinger-Reif, A. Evaluation of genetic risk score models in the presence of interaction and linkage disequilibrium. *Front. Genet.* **4**, 138. https://doi.org/10.3389/fgene.2013.00138 (2013).
23. Hüls, A. *et al.* Comparison of weighting approaches for genetic risk scores in gene–environment interaction studies. *BMC Genet.* **18**, 115. https://doi.org/10.1186/s12863-017-0586-3 (2017).
24. Lin, W.-Y. *et al.* Using genetic risk score approaches to infer whether an environmental factor attenuates or exacerbates the adverse influence of a candidate gene. *Front. Genet.* **11**, 331. https://doi.org/10.3389/fgene.2020.00331 (2020).
25. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **67**, 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x (2005).
26. Mavaddat, N. *et al.* Polygenic risk scores for prediction of breast cancer and breast cancer subtypes. *Am. J. Hum. Genet.* **104**, 21–34. https://doi.org/10.1016/j.ajhg.2018.11.002 (2019).
27. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **58**, 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x (1996).
28. Hoerl, A. E. & Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **12**, 55–67. https://doi.org/10.1080/00401706.1970.10488634 (1970).
29. Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer Science & Business Media, New York, 2009).
30. Breiman, L., Friedman, J. H., Stone, C. J. & Olshen, R. A. *Classification and Regression Trees* (CRC Press, 1984).
31. R Core Team. *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, 2022).
32. Schikowski, T. *et al.* Long-term air pollution exposure and living close to busy roads are associated with COPD in women. *Respir. Res.* **6**, 152. https://doi.org/10.1186/1465-9921-6-152 (2005).
33. Friedman, J., Hastie, T. & Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **33**, 1–22. https://doi.org/10.18637/jss.v033.i01 (2010).
34. Wright, M. N. & Ziegler, A. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *J. Stat. Softw.* **77**, 1–17. https://doi.org/10.18637/jss.v077.i01 (2017).
35. Malley, J. D., Kruppa, J., Dasgupta, A., Malley, K. G. & Ziegler, A. Probability machines: Consistent probability estimation using nonparametric learning machines. *Methods Inf. Med.* **51**, 74–81. https://doi.org/10.3414/ME00-01-0052 (2012).
36. McCarthy, S. *et al.* A reference panel of 64,976 haplotypes for genotype imputation. *Nat. Genet.* **48**, 1279–1283. https://doi.org/10.1038/ng.3643 (2016).
37. Das, S. *et al.* Next-generation genotype imputation service and methods. *Nat. Genet.* **48**, 1284–1287. https://doi.org/10.1038/ng.3656 (2016).
38. Reed, E. *et al.* A guide to genome-wide association analysis and post-analytic interrogation. *Stat. Med.* **34**, 3769–3792. https://doi.org/10.1002/sim.6605 (2015).
39. Beelen, R. *et al.* Effects of long-term exposure to air pollution on natural-cause mortality: An analysis of 22 European cohorts within the multicentre escape project. *The Lancet* **383**, 785–795. https://doi.org/10.1016/S0140-6736(13)62158-3 (2014).
40. Eeftens, M. *et al.* Development of land use regression models for pm2.5, pm2.5 absorbance, pm10 and pmcoarse in 20 European study areas; results of the escape project. *Environ. Sci. Technol.* **46**, 11195–11205. https://doi.org/10.1021/es301948k (2012).
41. Vanhoof, J., Declerck, K. & Geusens, P. Prevalence of rheumatic diseases in a rheumatological outpatient practice. *Ann. Rheum. Dis.* **61**, 453–455. https://doi.org/10.1136/ard.61.5.453 (2002).
42. Jokar, M. & Jokar, M. Prevalence of inflammatory rheumatic diseases in a rheumatologic outpatient clinic: Analysis of 12626 cases. *Rheumatol. Res.* **3**, 21–27. https://doi.org/10.22631/rr.2017.69997.1037 (2018).
43. Sangha, O. Epidemiology of rheumatic diseases. *Rheumatology* **39**, 3–12. https://doi.org/10.1093/rheumatology/39.suppl_2.3 (2000).
44. Canela-Xandri, O., Rawlik, K. & Tenesa, A. An atlas of genetic associations in UK Biobank. *Nat. Genet.* **50**, 1593–1599. https://doi.org/10.1038/s41588-018-0248-z (2018).

45. Bycroft, C. *et al.* The UK Biobank resource with deep phenotyping and genomic data. *Nature* **562**, 203–209. https://doi.org/10.1038/s41586-018-0579-z (2018).
46. Kampstra, A. S. & Toes, R. E. HLA class II and rheumatoid arthritis: The bumpy road of revelation. *Immunogenetics* **69**, 597–603. https://doi.org/10.1007/s00251-017-0987-5 (2017).
47. Clarke, A. & Vyse, T. J. Genetics of rheumatic disease. *Arthr. Res. Therapy* **11**, 1–9. https://doi.org/10.1186/ar2781 (2009).
48. Eyre, S. *et al.* High-density genetic mapping identifies new susceptibility loci for rheumatoid arthritis. *Nat. Genet.* **44**, 1336–1340. https://doi.org/10.1038/ng.2462 (2012).
49. Raychaudhuri, S. *et al.* Five amino acids in three HLA proteins explain most of the association between MHC and seropositive rheumatoid arthritis. *Nat. Genet.* **44**, 291–296. https://doi.org/10.1038/ng.1076 (2012).
50. Jiang, L., Jiang, D., Han, Y., Shi, X. & Ren, C. Association of HLA-DPB1 polymorphisms with rheumatoid arthritis: A systemic review and meta-analysis. *Int. J. Surg.* **52**, 98–104. https://doi.org/10.1016/j.ijsu.2018.01.046 (2018).
51. Okada, Y. *et al.* Contribution of a non-classical HLA gene, HLA-DOA, to the risk of rheumatoid arthritis. *Am. J. Hum. Genet.* **99**, 366–374. https://doi.org/10.1016/j.ajhg.2016.06.019 (2016).
52. Purcell, S. *et al.* PLINK: A tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* **81**, 559–575. https://doi.org/10.1086/519795 (2007).
53. Chang, C. C. *et al.* Second-generation PLINK: Rising to the challenge of larger and richer datasets. *GigaScience* **4**, 7. https://doi.org/10.1186/s13742-015-0047-8 (2015).
54. Källberg, H. *et al.* Gene-gene and gene–environment interactions involving HLA-DRB1, PTPN22, and smoking in two subsets of rheumatoid arthritis. *Am. J. Hum. Genet.* **80**, 867–875. https://doi.org/10.1086/516736 (2007).
55. Karlson, E. W. & Deane, K. Environmental and gene–environment interactions and risk of rheumatoid arthritis. *Rheum. Dis. Clin.* **38**, 405–426. https://doi.org/10.1016/j.rdc.2012.04.002 (2012).
56. Hüls, A. *et al.* Nonatopic eczema in elderly women: Effect of air pollution and genes. *J. Allergy Clin. Immunol.* **143**, 378–385. https://doi.org/10.1016/j.jaci.2018.09.031 (2019).
57. Krämer, U. *et al.* Traffic-related air pollution and incident type 2 diabetes: Results from the SALIA cohort study. *Environ. Health Perspect.* **118**, 1273–1279. https://doi.org/10.1289/ehp.0901689 (2010).
58. Mentch, L. & Zhou, S. Randomization as regularization: A degrees of freedom explanation for random forest success. *J. Mach. Learn. Res.* **21**, 1–36 (2020).
59. Janitza, S. & Hornung, R. On the overestimation of random forest's out-of-bag error. *PLoS ONE* **13**, 1–31. https://doi.org/10.1371/journal.pone.0201904 (2018).
60. Mitchell, M. W. Bias of the random forest out-of-bag (OOB) error for certain input parameters. *Open J. Stat.* **1**, 205–211. https://doi.org/10.4236/ojs.2011.13024 (2011).

Acknowledgements

Computational infrastructure and support were provided by the Centre for Information and Media Technology at the Heinrich Heine University Düsseldorf.

Author contributions

M.L. and H.S. developed the new GxE interaction testing procedures and designed the simulation study. M.L., S.K., and T.S. conceived the analyses of the real data application. The simulation study and real data evaluations were conducted by M.L. M.L. was the major contributor in writing the manuscript. All authors read and approved the final manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL. This work has been supported by the Research Training Group "Biostatistical Methods for High-Dimensional Data in Toxicology" (RTG 2624, Project R3) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation—Project Number 427806116). The SALIA follow-up study 2007–2010 was funded by the German Statutory Accident Insurance (DGUV) Grant No: 617.0-FP266 and the European Community's Seventh Framework Programme (FP 7/2007-2011) under the grant agreement number 211250.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-023-28172-4.

Correspondence and requests for materials should be addressed to M.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Additional file 1

Efficient gene-environment interaction testing through bootstrap aggregating

Michael Lau[1,2,*], Sara Kress[2], Tamara Schikowski[2] and Holger Schwender[1]

[1] *Mathematical Institute, Heinrich Heine University, Düsseldorf, Germany*
[2] *IUF – Leibniz Research Institute for Environmental Medicine, Düsseldorf, Germany*
[*] *Correspondence: michael.lau@hhu.de*

rsID	CHROM	POS	REF	ALT	MAF	TYPE	R2	ER2
rs9267989	6	32219320	G	T	0.178	IMPUTED	0.999	
rs9268145	6	32257284	T	G	0.194	GENOTYPED	1.000	0.995
rs522254	6	32273060	A	G	0.194	IMPUTED	0.994	
rs6910071	6	32282854	A	G	0.194	GENOTYPED	0.999	0.982
rs28361060	6	32303848	G	A	0.193	GENOTYPED	0.999	0.964
rs9268362	6	32333341	A	G	0.193	GENOTYPED	0.998	0.965
rs2073044	6	32338986	C	T	0.253	GENOTYPED	0.993	0.944
rs9268433	6	32345891	T	G	0.208	IMPUTED	0.983	
rs9268451	6	32349086	T	C	0.208	IMPUTED	0.983	
rs9268455	6	32349772	C	T	0.208	IMPUTED	0.983	
rs3793127	6	32371915	C	T	0.209	GENOTYPED	1.000	1.000
rs3763309	6	32375973	C	A	0.209	GENOTYPED	1.000	1.000
rs3763312	6	32376348	G	A	0.209	IMPUTED	0.999	
rs9268515	6	32379295	G	C	0.169	GENOTYPED	1.000	0.985
rs9268521	6	32381374	G	C	0.220	IMPUTED	0.996	
rs9268522	6	32381443	A	T	0.221	IMPUTED	0.995	
rs9268543	6	32384801	A	T	0.154	IMPUTED	0.997	
rs2395163	6	32387809	T	C	0.207	GENOTYPED	1.000	0.994
rs9268581	6	32396930	G	A	0.206	IMPUTED	0.996	
rs9268614	6	32402778	T	G	0.206	GENOTYPED	1.000	1.000
rs2395175	6	32405026	G	A	0.153	GENOTYPED	1.000	0.993
rs9268627	6	32405821	T	C	0.206	IMPUTED	0.999	
rs9268926	6	32433067	A	G	0.185	IMPUTED	0.985	
rs369515426	6	32542282	T	G	0.182	IMPUTED	0.683	
rs113322920	6	32553849	T	C	0.152	IMPUTED	0.794	
rs34855541	6	32559825	A	G	0.142	GENOTYPED	1.000	0.995
rs36096565	6	32560025	A	G	0.159	IMPUTED	0.850	
rs35395738	6	32560209	T	C	0.144	IMPUTED	0.900	
rs34415150	6	32560477	A	G	0.137	IMPUTED	0.849	
rs35118762	6	32560631	C	T	0.142	IMPUTED	0.986	
rs34928543	6	32560695	G	C	0.142	IMPUTED	0.987	
rs35265698	6	32561334	C	G	0.142	IMPUTED	0.984	
rs34350244	6	32561465	C	T	0.142	IMPUTED	0.944	
rs35294087	6	32561466	A	G	0.142	IMPUTED	0.977	
rs34553045	6	32561565	T	C	0.142	IMPUTED	0.941	
rs35371668	6	32561638	C	T	0.139	IMPUTED	0.889	
rs34647096	6	32561681	G	A	0.142	IMPUTED	0.977	
rs188575117	6	32561935	A	C	0.142	IMPUTED	0.923	
rs2760985	6	32566398	G	A	0.142	IMPUTED	0.978	
rs687308	6	32567256	C	T	0.142	IMPUTED	0.980	
rs35117964	6	32568146	A	G	0.151	IMPUTED	0.924	
rs34039593	6	32570311	T	G	0.142	GENOTYPED	1.000	1.000
rs2647066	6	32571122	C	T	0.142	IMPUTED	0.973	

rs17425622	6	32571961	T	C	0.142	IMPUTED	0.950	
rs601945	6	32573415	A	G	0.142	IMPUTED	0.967	
rs602457	6	32573562	T	C	0.142	IMPUTED	0.939	
rs7760841	6	32574868	C	T	0.138	IMPUTED	0.944	
rs560530	6	32577222	G	A	0.183	IMPUTED	0.984	
rs660895	6	32577380	A	G	0.183	GENOTYPED	1.000	0.999
rs532965	6	32577973	T	G	0.142	IMPUTED	0.997	
rs3997868	6	32578590	A	G	0.183	IMPUTED	0.992	
rs3997872	6	32580617	T	A	0.142	IMPUTED	0.996	
rs521539	6	32581973	G	A	0.183	GENOTYPED	1.000	1.000
rs3129751	6	32582189	A	C	0.142	IMPUTED	0.999	
rs3104415	6	32582577	A	C	0.324	IMPUTED	0.998	
rs34656207	6	32582601	C	T	0.346	IMPUTED	0.826	
rs3104413	6	32582650	C	G	0.142	GENOTYPED	0.999	0.972
rs3129754	6	32583046	A	G	0.408	IMPUTED	0.948	
rs3129756	6	32583063	A	G	0.376	IMPUTED	0.801	
rs6605556	6	32583099	A	G	0.142	IMPUTED	0.939	
rs4959106	6	32583159	T	C	0.430	GENOTYPED	1.000	0.996
rs6931044	6	32583194	G	T	0.408	IMPUTED	0.997	
rs34850435	6	32583299	C	T	0.431	IMPUTED	0.991	
rs6931277	6	32583357	A	T	0.142	GENOTYPED	1.000	0.993
rs6941972	6	32583529	G	A	0.032	IMPUTED	0.455	
rs36124427	6	32583677	T	C	0.430	IMPUTED	0.997	
rs1281935	6	32583820	G	T	0.050	IMPUTED	0.974	
rs34028938	6	32584346	C	A	0.430	IMPUTED	0.990	
rs510205	6	32584693	C	G	0.183	IMPUTED	0.985	
rs1281931	6	32587966	T	C	0.051	GENOTYPED	0.996	0.851
rs9271608	6	32591588	A	G	0.142	IMPUTED	0.989	
rs3104375	6	32600101	G	C	0.142	IMPUTED	0.953	
rs1391371	6	32603798	A	T	0.155	IMPUTED	0.917	
rs9272417	6	32605078	A	G	0.143	IMPUTED	0.934	
rs9272461	6	32605609	G	A	0.149	IMPUTED	0.951	
rs41269945	6	32607853	A	T	0.140	IMPUTED	0.873	
rs17426593	6	32608077	T	C	0.142	IMPUTED	0.873	
rs41269955	6	32608269	G	A	0.137	IMPUTED	0.854	
rs34141382	6	32608478	T	C	0.130	IMPUTED	0.848	
rs34763586	6	32608998	T	C	0.140	IMPUTED	0.862	
rs34965214	6	32609545	C	T	0.142	IMPUTED	0.834	
rs9272785	6	32610401	G	A	0.142	IMPUTED	0.844	
rs28724243	6	32629347	T	C	0.305	IMPUTED	0.850	
rs9275222	6	32659516	A	T	0.461	IMPUTED	0.988	
rs4713582	6	32660051	T	C	0.496	GENOTYPED	1.000	1.000
rs9275511	6	32674329	G	A	0.456	IMPUTED	0.935	
rs7764856	6	32680640	T	A	0.326	GENOTYPED	0.999	0.984

Table S1: Information on the 87 SNPs analyzed in the real data application using the SALIA data set. rsID: Reference SNP cluster ID; CHROM: Chromosome; POS: Reference position; REF: Reference allele; ALT: Alternative non-reference allele; MAF: Minor allele frequency; TYPE: Variant genotyped or imputed; R2: Imputation quality (estimate of the squared correlation between imputed genotypes and true/unobserved genotypes); ER2: Empirical R^2 for genotyped variants (not calculated for imputed variants).

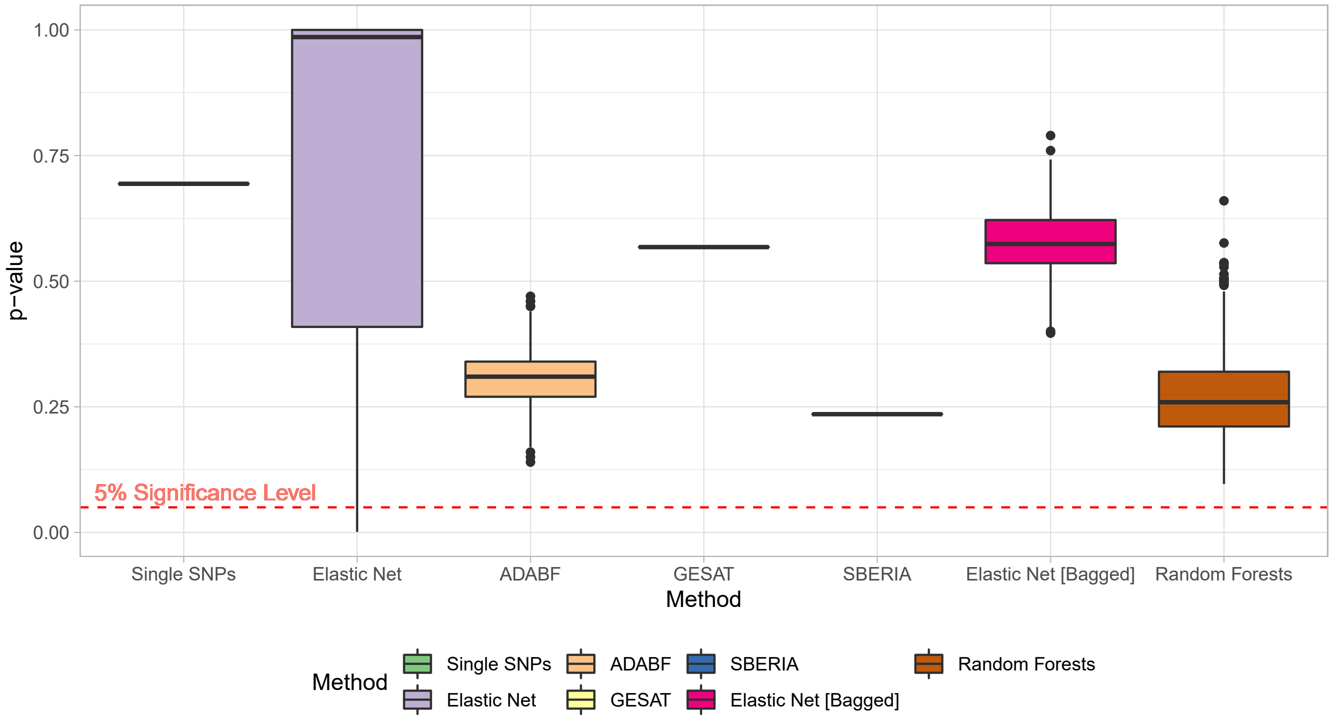

Figure S1: p-values of 1000 independent applications of the GxE interaction testing procedures to the considered real data set from the SALIA cohort study analyzing the gene-based SNP selection containing 72 SNPs. For elastic net, the train/test data splits changed. For the two bagging-based tests, the bootstrap samples changed. For ADABF, the random sampling from the null distribution of GxE interaction coefficients changed. The single-SNP-based test, GESAT, and SBERIA were applied only once, since there is no randomness involved in the application of these tests.

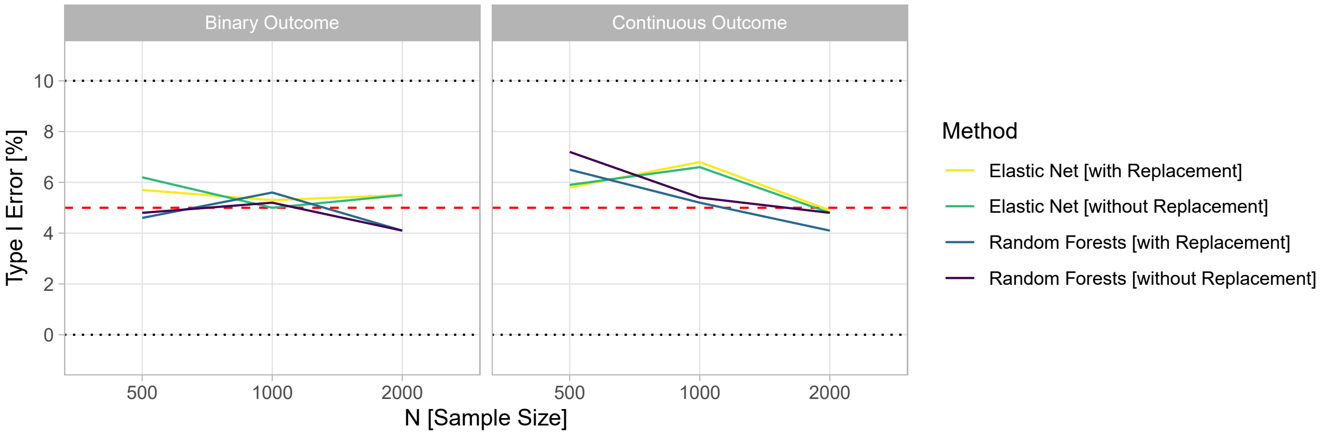

Figure S2: Type I error rates of the bagging-based GxE interaction tests using sampling with or without replacement in the simulation study

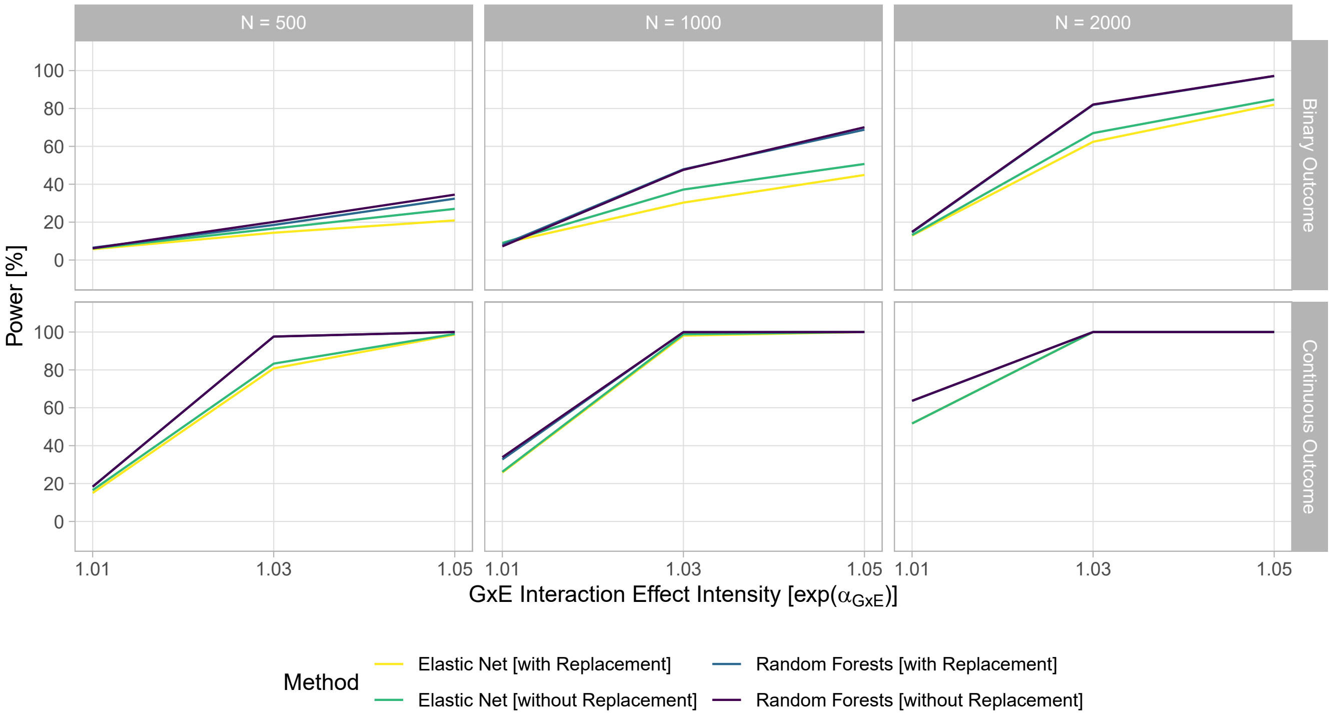

Figure S3: Power of the bagging-based GxE interaction tests using sampling with or without replacement in the simulation study

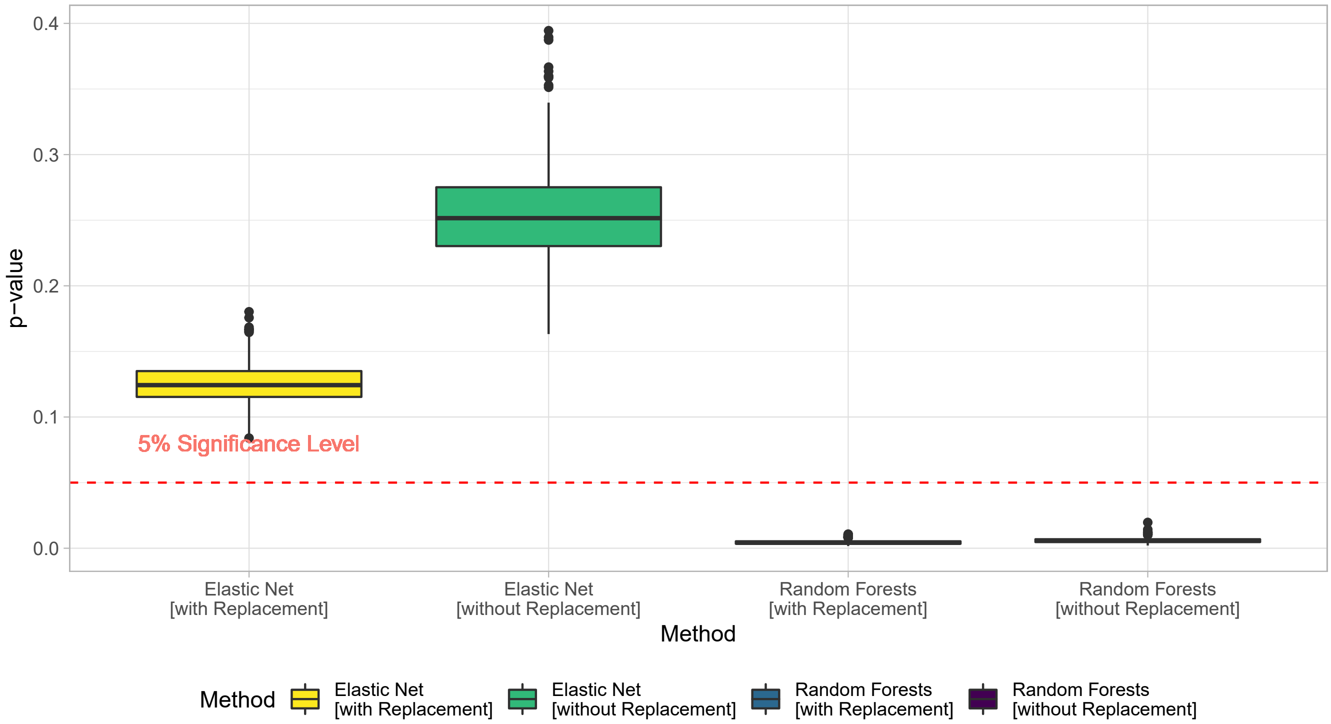

Figure S4: p-values of the bagging-based GxE interaction tests using sampling with or without replacement in the real data application

Chapter 4

logicDT: a procedure for identifying response-associated interactions between binary predictors

In the following, the third manuscript [Lau et al., 2024], which was published in the journal Machine Learning, is presented and addresses Research Gaps 1 and 4–6.

logicDT: a procedure for identifying response-associated interactions between binary predictors

Michael Lau, Tamara Schikowski, and Holger Schwender

Authorship:	First author
Contributed part:	85%
Contribution:	Development of the statistical methods Mathematically proving theoretical properties Implementation in software Designing the simulation study Conceiving the real data analyses Conducting the statistical data analyses Preparing figures and tables Interpretation of results Writing the paper
Journal:	Machine Learning
Impact factor:	7.454 (2022)
Date of publication:	22 December 2023
DOI:	10.1007/s10994-023-06488-6

Machine Learning (2024) 113:933–992
https://doi.org/10.1007/s10994-023-06488-6

logicDT: a procedure for identifying response-associated interactions between binary predictors

Michael Lau[1,2] · Tamara Schikowski[2] · Holger Schwender[1]

Received: 3 November 2022 / Revised: 13 October 2023 / Accepted: 16 November 2023 /
Published online: 22 December 2023

Abstract

Interactions between predictors play an important role in many applications. Popular and successful tree-based supervised learning methods such as random forests or logic regression can incorporate interactions associated with the considered outcome without specifying which variables might interact. Nonetheless, these algorithms suffer from certain drawbacks such as limited interpretability of model predictions and difficulties with negligible marginal effects in the case of random forests or not being able to incorporate interactions with continuous variables, being restricted to additive structures between Boolean terms, and not directly considering conjunctions that reveal the interactions in the case of logic regression. We, therefore, propose a novel method called logic decision trees (logicDT) that is specifically tailored to binary input data and helps to overcome the drawbacks of existing methods. The main idea consists of considering sets of Boolean conjunctions, using these terms as input variables for decision trees, and searching for the best performing model. logicDT is also accompanied by a framework for estimating the importance of identified terms, i.e., input variables and interactions between input variables. This new method is compared to other popular statistical learning algorithms in simulations and real data applications. As these evaluations show, logicDT is able to yield high prediction performances while maintaining interpretability.

Keywords Decision trees · Interpretable machine learning · Regression procedure · Variable importance measures · Importance of interactions · Polygenic risk scores

Editor: Johannes Fürnkranz.

✉ Michael Lau
michael.lau@hhu.de

Tamara Schikowski
tamara.schikowski@iuf-duesseldorf.de

Holger Schwender
holger.schwender@hhu.de

[1] Mathematical Institute, Heinrich Heine University, Universitätsstrasse 1, 40225 Düsseldorf, Germany

[2] IUF - Leibniz Research Institute for Environmental Medicine, Auf'm Hennekamp 50, 40225 Düsseldorf, Germany

1 Introduction

In many practically relevant applications, a proper coverage of interactions between predictors is key for constructing strong predictive models. One particularly important example is the analysis of genetic or environmental risk factors in epidemiological and medical studies for, e.g., constructing genetic/polygenic risk scores (Che & Motsinger-Reif, 2013; Ho et al., 2019) that can be viewed as a function $\varphi : \mathcal{X} \rightarrow \mathcal{Y}$ from the p-dimensional space $\mathcal{X} = \{0, 1, 2\}^p$ of p SNPs (single nucleotide polymorphisms), i.e., single base-pair substitutions in the DNA, to the response space $\mathcal{Y}$ assigning a risk estimate. For example, for a binary outcome such as a binary disease status, a probability estimate $\hat{P}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \in [0, 1]$ of developing this disease might be a proper risk estimate. Since SNPs are variables with three possible outcomes counting the number of minor allele occurrences with respect to both chromosomes, i.e., how often the less frequent variant occurs in an individual, they can be easily (and biologically meaningful) divided into two binary variables each, i.e., in $\text{SNP}_D = \mathbb{1}(\text{SNP} \neq 0)$ and $\text{SNP}_R = \mathbb{1}(\text{SNP} = 2)$, coding for a dominant and a recessive effect, respectively. It is well-known that in the analysis of genetic features such as SNPs, interactions, e.g., gene-gene interactions (Che & Motsinger-Reif, 2013) and gene-environment interactions (Ottman, 1996), play a crucial role. Especially in this setting, not only a high predictive ability of the resulting models, but also a high interpretability for understanding which and how genetic variants influence the risk of disease is desirable.

Tree-based statistical learning methods such as decision trees, random forests, or logic regression are very popular and versatile in recognizing underlying data structures. These methods have been already applied to analyze SNP data (e.g., Bureau et al., 2005; Winham et al., 2012; Ruczinski et al., 2004). However, these methods typically fail at simultaneously achieving a reliable predictive strength and a high interpretability of how exactly predictions are composed.

In this article, we propose the tree-based supervised learning procedure *logicDT* (logic decision trees) which is specifically tailored for properly incorporating interactions between binary predictors. Continuous relationships of additional covariates and interactions of these covariates with the binary variables can also be covered by this procedure. logicDT is designed for yielding highly interpretable prediction models, while maintaining a high predictive ability. For measuring the influence of predictors and their interactions, a novel variable importance measure framework is proposed which, in principle, can be used in conjunction with any other learning procedure.

We start with briefly discussing similar methods and efforts on enhancing existing algorithms in Sect. 2. Then, logicDT and its extensions are presented in detail in Sect. 3. We additionally prove that logicDT is consistent. In Sect. 4, the novel variable importance measuring framework for estimating the influence of input variables and their interactions is proposed. Empirical studies on simulated data as well as on real data follow in Sect. 5 illustrating logicDT's properties in practice and comparing logicDT to other procedures. Sections 6 and 7 contain discussions and concluding remarks.

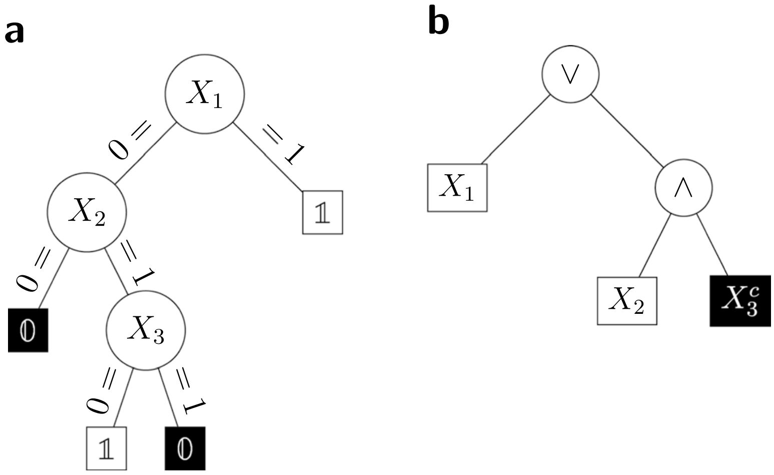

Fig. 1 Exemplary tree models for three binary input variables X_1, X_2 and X_3 predicting two different classes 0/false and 1/true. In **a**, a classification tree is shown. **b** depicts a logic tree describing the Boolean expression $X_1 \vee (X_2 \wedge X_3^c)$, where negations are denoted by ^{c} in this article. For the logic tree, terminal nodes containing negated predictors are depicted as black squares containing white text. Vice versa, non-negated predictors are depicted as white squares containing black text. Both trees are equivalent, i.e., they perform the same predictions for each predictor setting. Adapted from Lau et al. (2022)

2 Background and related work

In the following, we briefly discuss tree-based supervised learning procedures and their extensions.

2.1 Decision trees and random forests

One very popular and powerful statistical learning method are decision trees. Important implementations include classification and regression trees (CART) (Breiman et al., 1984) and C4.5 (Quinlan, 1993). Decision trees recursively partition the predictor space $\mathcal{X}$ considering one predictor per split into disjoint patches, to which individually a prediction value will be assigned. For predicting new outcomes, one starts at the root node and follows the edges corresponding to the specific predictor setting until a leaf is reached. Figure 1a illustrates an exemplary decision tree consisting of three binary predictors in a binary classification scenario.

Algorithm 1: Decision Tree Fitting

```
 1 function fitDecisionTree(Training data $\mathcal{D}$):
 2     Create an empty decision tree $T$ with root node $t_0$
 3     Initialize an empty stack where each element is a tuple $(t, \mathcal{D}_t)$
 4     stack.push($(t_0, \mathcal{D})$)
 5     while $|stack| > 0$ do
 6         $(t, \mathcal{D}_t) =$ stack.pop()
 7         if Stopping criterion is met then
 8             $\widehat{Y}_t = \frac{1}{|\mathcal{D}_t|} \sum_{(x,y) \in \mathcal{D}_t} y$
 9         else
10             splits = Initialize empty list
11             for every input variable $X_j$ do
12                 splits.append(Best split on $X_j$)
13             end
14             $s^* =$ Best split in splits
15             Split the inner node $t$ in $T$ on $s^* = (\mathcal{X}_{t_L}, \mathcal{X}_{t_R})$
16             stack.push($(t_L, \mathcal{D}_{t_L})$)
17             stack.push($(t_R, \mathcal{D}_{t_R})$)
18         end
19     end
20     return $T$
21 end
```

Similar to Louppe (2014), Algorithm 1 summarizes the fitting process of decision trees. In Lines 11 through 14, the locally best split, i.e., the predictor and the splitting point which maximize the node homogeneity after splitting is identified and used for further splitting the tree into two subnodes. For measuring the homogeneity, an impurity measure i is used which assigns a node an estimate of its heterogeneity. For evaluating the strength of a split s partitioning the node t into two child nodes t_L and t_R, the impurity reduction

$$\Delta i(s,t) \ := \ i(t) - \frac{n_{t_L}}{n_t} i(t_L) - \frac{n_{t_R}}{n_t} i(t_R) \ \geq \ 0 \tag{1}$$

for the number of training observations n_t falling into node t is maximized. For regression purposes, the impurity measure of the mean squared error

$$i_{\text{Regression}}(t) \ := \ \frac{1}{n_t} \sum_{(\boldsymbol{x},y) \in \mathcal{D}_t} (y - \hat{y}_t)^2 \tag{2}$$

is used as the impurity measure considering the subset $\mathcal{D}_t$ of the training data set $\mathcal{D}$ to node t and the predicted outcome $\hat{y}_t$ in node t. For classification or risk estimation, the Gini impurity

$$i_{\text{Gini}}(t) \ := \ \sum_{c \in \mathcal{Y}} \frac{n_{c,t}}{n_t} \left(1 - \frac{n_{c,t}}{n_t}\right) \tag{3}$$

is used for classes $c \in \mathcal{Y}$ and their corresponding frequency $n_{c,t}$ in node t. An alternative popular impurity measure for classification tasks is the information gain

$$i_{\text{Entropy}}(t) \ := \ -\sum_{c\in\mathcal{Y}} \frac{n_{c,t}}{n_t} \log_2\left(\frac{n_{c,t}}{n_t}\right)$$

that is based on the Shannon entropy (e.g., Louppe, 2014).

The partitioning of a tree branch locally stops when the training data cannot be further divided, i.e., if for all $(\boldsymbol{x}, y), (\boldsymbol{x}', y') \in \mathcal{D}_t$, it either holds $\boldsymbol{x} = \boldsymbol{x}'$ or $y = y'$ (see Line 7 of Algorithm 1). Usually, to prevent overfitting, additional stopping criteria are used such as the minimum node size, i.e., the minimum number of training observations falling into a leaf, or a minimum impurity reduction which has to be achieved in order to split the node. However, these additional stopping criteria yield hyperparameters which, thus, require proper tuning. Finally, the last important step is the assignment of a predicted value to a leaf (Line 8 of Algorithm 1). Although theoretically, this predicted value is already used for evaluating the splits. The prediction values are obtained by empirical risk minimization yielding the arithmetic mean for regression tasks. For binary risk estimation, also the arithmetic mean of the outcome Y given the predictor values $\boldsymbol{x}$ is used if Y is coded as 0 or 1. If pure classifications are considered, the class with the lowest risk estimate is chosen.

A particularly popular and successful extension of decision trees are random forests which build ensembles of randomized decision trees yielding even higher predictive performance at the cost of losing interpretability of the fitted models (Breiman, 2001). The randomization is performed by employing bagging (Breiman, 1996), which is described in more detail in Sect. 3.9, and by considering random predictor subsets for splitting at each node. Random forests can substantially outperform single decision trees due to the instability issue of decision trees, which states that small noise-like changes of the training data set can lead to large modifications of the fitted model. This instability issue is mainly caused by the greedy fashion of choosing splits (Li & Belford, 2002; Murthy & Salzberg, 1995).

If deep trees are grown, both single decision trees and random forests can overfit (Hastie et al., 2009; Tang et al., 2018). For certain, not necessarily realistic scenarios (e.g., no subsampling combined with totally randomized trees in which the splits are chosen independent of the outcome or too extreme subsampling in which the subsample size remains constant, but the sample size approaches infinity), Tang et al. (2018) proved that random forests with deeply grown trees are inconsistent.

If shallow trees are grown, fruitful splits might be left out. Furthermore, decision trees and random forests struggle uncovering interactions effects, if the interacting variables only exhibit negligible marginal effects (Wright et al., 2016). Moreover, due to the prediction values of the leaves being constant for finitely many predictor scenarios in conventional decision trees and random forests, continuous function relationships can only be approximated by step functions. However, for example, in the analysis of genetic and environmental risk factors of certain diseases, in which random forests are frequently used (Winham et al., 2012; Bellinger et al., 2017), a continuous influence of an environmental factor on the disease risk is reasonable.

There are a variety of modifications to decision trees and random forests which try to overcome the issues mentioned above. These methods, however, address individual issues. In the following section, we will discuss some of these modifications.

2.2 Extensions of decision trees and random forests

For improving the ability on detecting interactions, one well-known approach is the usage of multivariate splits, i.e., splits based on multiple variables at once, e.g., by using linear combinations of the predictors. Exemplary methods of this class are oblique decision trees (Murthy et al., 1994) and oblique random forests (Menze et al., 2011), where a particular implementation of the latter is, e.g., SPORF (Sparse Projection Oblique Randomer Forests; Tomita et al., 2020). For binary predictors as considered in this article, these multivariate linear splits can be used for creating Boolean conjunctions of predictors, thus, potentially splitting on an interaction. However, methods that try to linearly separate the current feature space based on the (binary) class label in each splitting node (such as the method proposed by Menze et al., 2011) are only suited to classification tasks. Another recent modification is interaction forests (Hornung & Boulesteix, 2022) which directly searches for interaction splits at each node. An overview over such interaction-focused modifications of decision trees and random forests is, e.g., given by Hornung and Boulesteix (2022).

The greedy search algorithm employed in classic decision tree fitting procedures (such as in CART) is fast and scales to high-dimensional problems. However, as the greedy search conducts local searches for splits, it requires detectable marginal effects to identify interaction effects. For example, if X_1 and X_2 interact with each other, X_1 or X_2 have to be individually identified first as splitting variables. Due to increasing computational capabilities, optimal decision trees have been proposed by Nijssen and Fromont (2010) and Bertsimas and Dunn (2017) to perform a global optimization. In the former method, namely DL8 (decision trees from lattices), dynamic programming is utilized to fit decision trees. In the latter method, namely OCT (optimal classification trees), the decision tree fitting problem is phrased as a mixed-integer optimization problem. More recently, alternative optimal decision tree algorithms that utilize dynamic programming such as DL8.5 (Aglin et al., 2020a) and MurTree (Demirović et al., 2022) and optimal decision tree fitting procedures that incorporate multivariate splits such as WODT (Yang et al., 2019) and SVM1-ODT (Zhu et al., 2020) have been proposed. A review of optimal decision tree fitting procedures is, e.g., given by Carrizosa et al. (2021).

Blockeel and De Raedt (1998) proposed combining decision trees with logic programming. Their method is called TILDE (top-down induction of logical decision trees). At each inner node, a Boolean conjunction is responsible for further partitioning the input data. Model fitting is performed in a greedy fashion very similar as in C4.5 (Quinlan, 1993). However, the space of eligible splits, over which the greedy search is applied, has to be defined by the user by utilizing background knowledge and, e.g., specifying which variables may be part of the same conjunction. Another important difference between TILDE and other decision tree algorithms is that TILDE uses logic programs for specifying data examples. This is in contrast to the statistical learning setup considered in this article. We consider the standard setting, in which data are given in a tabular format and relevant background knowledge about the relationships of certain variables is not available.

Rule extraction methods aim at increasing the interpretability of tree ensemble methods while keeping their predictive strength. They start by fitting a tree ensemble such as random forests and try to extract the most important prediction rules from the individual decision tree paths. These prediction rules are then gathered in rule lists yielding the final model, in which predictions are made according to which rules hold true. One of the first and most established rule extraction methods is RuleFit (Friedman & Popescu, 2008), which fits a boosted ensemble of decision trees and selects the most important rules using the lasso (Tibshirani, 1996). Alternative rule extraction methods include node harvest (Meinshausen, 2010) and SIRUS (Stable and Interpretable Rule Set, Bénard et al., 2021), which both fit random forests for generating the models from which the rules are to be extracted.

For modeling continuous regression models in the leaves, typically, GLMs are employed such as in MOB (model-based recursive partitioning, Zeileis et al., 2008). An overview on several GLM-based approaches is, e.g., given by Rusch and Zeileis (2013). However, the right parametric model might not be known prior to fitting models so that a more flexible non-linear regression model might be preferable. Moreover, these methods do not lay a focus on properly handling interactions between the splitting variables.

2.3 Logic regression

Logic regression (Ruczinski et al., 2003) is another tree-based supervised learning method. It has been specifically developed for analyzing SNP data and is, therefore, frequently used in such analyses (e.g., Ruczinski et al., 2004; Zhi et al., 2015). Logic regression is focussed on binary predictors and tries to identify Boolean combinations of the predictors that shall explain the variation in the outcome. These Boolean expressions can also be presented as logic trees, i.e., trees holding predictors (or their negations) in their leaves and recursively combining them with the Boolean AND-operator (denoted by $\wedge$ in the following) or the Boolean OR-operator (denoted by $\vee$ in the following) using inner nodes. Figure 1b illustrates an exemplary logic tree corresponding to the Boolean expression $X_1 \vee (X_2 \wedge X_3^c)$. If a true logic tree is identified with class 1 and a false logic tree is identified with class 0, this tree is equivalent to the classification tree from Fig. 1a.

To generalize the usage of logic regression to regression purposes, logic trees are embedded in GLMs, i.e., a model of the form

$$g(\mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]) = \beta_0 + \beta_1 L_1(\boldsymbol{x}) + \cdots + \beta_m L_m(\boldsymbol{x})$$

is considered for a link function g and logic trees $L_1, \ldots, L_m$. In general, every possible logic regression model can be transformed into an equivalent decision tree, and vice versa (Ruczinski et al., 2003). However, logic trees tend to be more sparse, i.e., by using Boolean logic, logic trees can describe the same prediction model with fewer nodes than decision trees in certain scenarios. For example, even in the simple prediction model depicted in Fig. 1, the logic tree consists of five nodes, whereas seven nodes are required in the CART tree to represent the Boolean expression. Note that this tree sparsity property holds true for binary classification scenarios in which a hard classification task instead of a more general class probability estimation task is considered.

The fitting procedure in logic regression is performed by a global stochastic search over all possible models, i.e., logic trees $L_1, \ldots, L_m$ and their GLM coefficients $\beta_0, \ldots, \beta_m$, where these GLM coefficients are determined by fitting a GLM using the considered logic trees as predictors in each step of the global stochastic search. In particular, simulated annealing (Kirkpatrick et al., 1983) is employed using simple modifications of the current model/state, i.e., adding or removing branches, exchanging variables or operators, and splitting or removing variables. Alternatively, a greedy local search always moving to the best neighbor state can be employed. However, this faster search comes without any guarantees of finding a globally optimal state. For evaluating the current state, a score function such as the mean squared error for linear regression or the deviance for logistic regression is used. For a detailed description and discussion of logic regression, see Ruczinski et al. (2003).

Single logic regression models tend to be unstable, if the signal is weak or if many predictors are actually predictive. One approach to tackle this problem is to apply bagging to logic regression models (Schwender & Ickstadt, 2007). However, similar to random forests, these models are no longer easily interpretable.

Even single logic regression models can be hard to interpret due to possibly complex logic tree structures. Typically, one is interested in the statistical interaction of predictors, which can be defined as the effect of the presence of certain predictor settings at once, i.e., using Boolean conjunctions, since conjunctions of input variables directly reveal the specific type of interaction that is considered (Chen et al., 2011). By De Morgan's laws, if a Boolean disjunction needs to be represented, the negation of the conjunction containing the negations of the input terms can be used, i.e., making disjunctions obsolete if all negations are available.

Logic regression can only take quantitative covariables additively into account by adding them to the linear predictor of the GLM containing the logic trees as single terms. Thus, no interactions between the binary predictors and quantitative predictors can be included. Similarly, interactions between logic trees themselves can also not be captured, thus, relying on the additive structure of the individual terms. If, for example, the scale of an underlying linear predictor is unknown, being able to also model interactions between the terms can be beneficial. Consider, e.g., the regression function

$$\begin{aligned}\mathbb{E}[Y \mid \boldsymbol{X}] &= \left[\alpha \cdot \mathbb{1}(X_1) + \beta \cdot \mathbb{1}(X_2 \wedge X_3^c)\right]^2 \\ &= \alpha^2 \cdot \mathbb{1}(X_1) + 2 \cdot \alpha \cdot \beta \cdot \mathbb{1}(X_1 \wedge X_2 \wedge X_3^c) + \beta^2 \cdot \mathbb{1}(X_2 \wedge X_3^c).\end{aligned}$$

On the squared scale, the terms X_1 and $X_2 \wedge X_3^c$ do not interact. However, on the original scale, if both terms are true at once, the linear predictor is adjusted by an additional $2\alpha\beta$.

3 Logic decision trees

To overcome the issues mentioned in the last section, we propose a novel method, called *logicDT (logic decision trees)*, which combines decision trees and an improved version of the Boolean term search of logic regression.

We define logic decision trees to be decision trees that can use Boolean conjunctions of input variables as splitting variables, which is in contrast to standard decision tree procedures. Logic decision trees may be used for regression purposes, in which—similar to regression trees—each leaf holds a direct estimate of the outcome, or for classification purposes, in which—similar to probability estimation trees (Provost & Domingos, 2003; Malley et al., 2012)—each leaf holds an estimate of the class membership probability. As discussed in Sect. 3.5, logic decision trees may also contain regression models in their leaves for modeling continuous relationships.

Allowing Boolean conjunctions of input variables as splitting variables, firstly, simplifies the resulting decision tree. If we, e.g., consider an outcome that is only altered if $X_1^c \wedge X_2$ holds, then creating a tree stump (i.e., a decision tree consisting of only one split) splitting on $X_1^c \wedge X_2$ would be sufficient when using logicDT, whereas a common decision

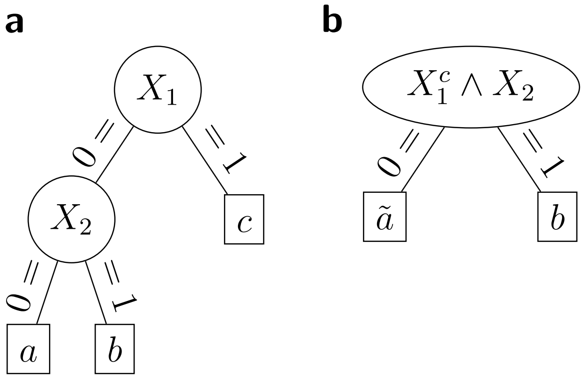

Fig. 2 Decision trees for splitting on $X_1^c \wedge X_2$. In **a**, a standard decision tree splitting on single input variables is shown. In **b**, a Boolean conjunction is used for splitting

tree only using single input variables for splitting would require a split on X_1 and another split on X_2 in the branch in which $X_1 = 0$ holds (see Fig. 2).

Secondly, this makes the prediction values in some leaves more robust. In our example, the common decision tree in Fig. 2a would further distinct between $X_1 = 1$ and $X_1^c \wedge X_2^c = 1$, while the tree in Fig. 2b uses one shared prediction, thus, utilizing more observations for creating the prediction value. Thirdly, due to the greedy search employed in standard decision tree splitting approaches, the interaction might not be found due to potentially negligible marginal effects of, in our example, X_1 or X_2 leading to splitting on other variables or not splitting at all, if a stopping criterion is triggered.

In the following subsections, logicDT is presented in detail.

3.1 Preliminaries

Let $\boldsymbol{X} = (X_1, \dots, X_p)$ be a p-dimensional random vector of binary input variables taking values in the p-dimensional space $\mathcal{X} = \{0, 1\}^p$ and let Y be a target random variable taking values in the space $\mathcal{Y}$. Let $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$ be a training data set with independent and identically distributed observations from the joint probability distribution of $(\boldsymbol{X}, Y)$. Then the corresponding statistical learning task can be formulated as estimating the true regressor $\mathbb{E}_{(\boldsymbol{X},Y)}[Y \mid \boldsymbol{X} = \cdot\,]$ by a function $\varphi : \mathcal{X} \to \mathcal{Y}$ using the training data set $\mathcal{D}$ (e.g., Hastie et al., 2009).

In this article, Boolean conjunctions between binary input variables are denoted using the Boolean $\wedge$ (AND) and negations of binary input variables are denoted using a superscript ^{c} (complement), i.e., $X_j^c = 1 - X_j$.

logicDT is aimed at identifying response-associated interactions, where two input variables X_i and X_j are defined to interact with each other with respect to the outcome Y, if the effect of one input variable (i.e., the partial derivative/finite differences of $\mathbb{E}[Y \mid \boldsymbol{X}]$ with respect to one input variable) depends on the other input variable (Sorokina et al., 2008). Therefore, if there is no interaction between X_i and X_j, the regression function $\mu(\boldsymbol{X}) = \mathbb{E}[Y \mid \boldsymbol{X}]$ can be decomposed into a sum $\mu(\boldsymbol{X}) = \mu_{\setminus i}(\boldsymbol{X}_{\setminus i}) + \mu_{\setminus j}(\boldsymbol{X}_{\setminus j})$, where $_{\setminus i}$ denotes leaving out the ith entry of the vector of input variables (Friedman & Popescu, 2008). This definition can be directly generalized to (statistical) interactions of arbitrary order. If there is no interaction between $X_{(1)}, \dots, X_{(k)}$, μ can be decomposed into a sum of functions, in which no summand is a function of all considered variables $X_{(1)}, \dots, X_{(k)}$ simultaneously.

In this article, we mainly focus on binary input variables. Therefore, every function $\varphi : \mathcal{X} \to \mathcal{Y}$ mapping from a p-dimensional space of binary input variables to a real number can be expressed as a sum of the form

$$\varphi(\boldsymbol{X}) = \beta_0 + \sum_{j=1}^{m} \beta_j \cdot \mathbb{1}\left(X_{k_{j,1}}^{(c)} \land \cdots \land X_{k_{j,p_j}}^{(c)}\right),$$

where $^{(c)}$ denotes potentially negating the considered variable and $k_{j,i}$ is the index of the ith variable in the jth summand. Hence, binary input variables $X_{(1)}, \dots, X_{(k)}$ interact with each other (with respect to Y), if μ cannot be decomposed without using a Boolean conjunction that simultaneously includes $X_{(1)}, \dots, X_{(k)}$. Boolean disjunctions are not considered in logicDT, since, by De Morgan's laws, Boolean disjunctions can be expressed using Boolean conjunctions and negations.

3.2 Core methodology of logicDT

The aim of logicDT is to identify important input variables and Boolean conjunctions of input variables to perform accurate predictions of the outcome. An input variable or a Boolean conjunction of input variables will be in the following referred to as a *term*. A set of terms will be referred to as a *state*. Examples of possible states would be

$$\{\{X_{73}\}\} \quad \text{or} \quad \{\{X_1^c \land X_2\}, \{X_5\}, \{X_9 \land X_{14}^c \land X_{42}^c\}\}.$$

In logicDT, states are obtained by a global stochastic search procedure that is introduced later in this section.

Logic decision trees are induced by identifying a state and exclusively using the terms contained in this state as input variables for fitting a conventional decision tree. For example, the three terms $X_1^c \land X_2$, X_5, and $X_9 \land X_{14}^c \land X_{42}^c$ are used as input variables to induce a decision tree, if the corresponding state $\{\{X_1^c \land X_2\}, \{X_5\}, \{X_9 \land X_{14}^c \land X_{42}^c\}\}$ is considered. Hence, creating a logic decision tree based upon a state is a two-stage procedure. First, the original training data set is transformed into a *tree training data set* using the terms of the considered state. Next, using this tree training data set, a decision tree is fitted.

For a set consisting of m terms

$$\left\{\left\{X_{k_{1,1}}^{(c)}, \dots, X_{k_{1,p_1}}^{(c)}\right\}, \dots, \left\{X_{k_{m,1}}^{(c)}, \dots, X_{k_{m,p_m}}^{(c)}\right\}\right\},$$

the original training data set is transformed into a tree training data set by constructing a $n \times (m+1)$ data matrix containing the m different predictors or conjunctions and the outcome. For example, if a training data set is given by

$$\mathcal{D} = \left[\begin{array}{c|c} \boldsymbol{x}_1 & y_1 \\ \boldsymbol{x}_2 & y_2 \\ \boldsymbol{x}_3 & y_3 \\ \boldsymbol{x}_4 & y_4 \end{array}\right] = \begin{array}{c} \begin{array}{cccc} X_1 & X_2 & X_3 & Y \end{array} \\ \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \end{array}$$

and the state $s = \{\{X_1\}, \{X_2 \land X_3^c\}\}$ is identified by the global stochastic search, the tree training data set, which is directly used for fitting the decision tree, is given by

$$\mathcal{D}_s = \left[\begin{array}{cc|c} X_1 & X_2 \wedge X_3^c & Y \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]. \tag{4}$$

Since each term is a binary variable itself, there is only one possible split of the data based on this term. Thus, the tree fitting procedure only needs to consider one split per input term, which makes the identification of the best local split particularly fast. For evaluating potential node splits and selecting the split, the conventional node impurity splitting criterion from Eq. (1) is used. For regression tasks, the MSE (mean squared error) impurity (see Eq. (2)) is used, and for classification tasks, the Gini impurity (see Eq. (3)) is used.

After the tree corresponding to the current state has been fitted, its performance on the training data is evaluated by passing all observations through the tree and calculating a *score* that measures the training data error, where the score is chosen so that a smaller value of the score corresponds to a better fit. For regression purposes, the MSE is employed. For risk estimation/classification purposes, probability estimation trees (Provost & Domingos, 2003; Malley et al., 2012) are grown that directly hold class probability estimates in their leaves by using empirical probabilities, i.e., using proportions of class occurrences. Thus, for scoring a state in the risk estimation/classification setting, the deviance is used, which is also known as the cross entropy or the negative binomial log-likelihood.

Alternatively, the negative area under the curve with respect to the receiver operating characteristic (AUC) might be used. However, the AUC does not capture the magnitude of the risk estimate in contrast to the deviance. Another alternative is the Brier score, which is the mean squared error between the risk estimate and the actual outcome.

For identifying an ideal state, logicDT performs a global search over all eligible states. The search is performed by using the current state to construct a decision tree, evaluating the performance of this tree, modifying the current state, and repeating this procedure. Modifications of logicDT states are called *neighbors* and are implicitly defined by slightly altering a given state. Figure 3 illustrates the possible state modifications/neighbor states using exemplary states. In the center of this figure, the current state is depicted. The possible state changes include

- exchanging or negating single variables (see, e.g., the replacement of X_2 by X_4 in the top and the negation of X_2 in the bottom of Fig. 3),
- adding or removing single variables from a term (see, e.g., the addition of X_8 in the top right and the removal of X_3^c in the bottom right of Fig. 3),
- adding or removing logic terms consisting of exactly one variable (see, e.g., the addition of X_{10} in the top left and the removal of X_2 in the bottom left of Fig. 3).

To avoid tautologies and uninformative terms, some specific alterations are prohibited. More precisely, the same variable should not occur more than once in a single term and the same term should not occur more than once in the proposed state.

The search is initialized by finding the single input variable that minimizes the score function, e.g., $\{\{X_{73}\}\}$. Using this initial state, a global optimization procedure employing simulated annealing (Kirkpatrick et al., 1983) is carried out for finding the state that minimizes the score function, i.e., now permitting all possible states potentially consisting of more than one term.

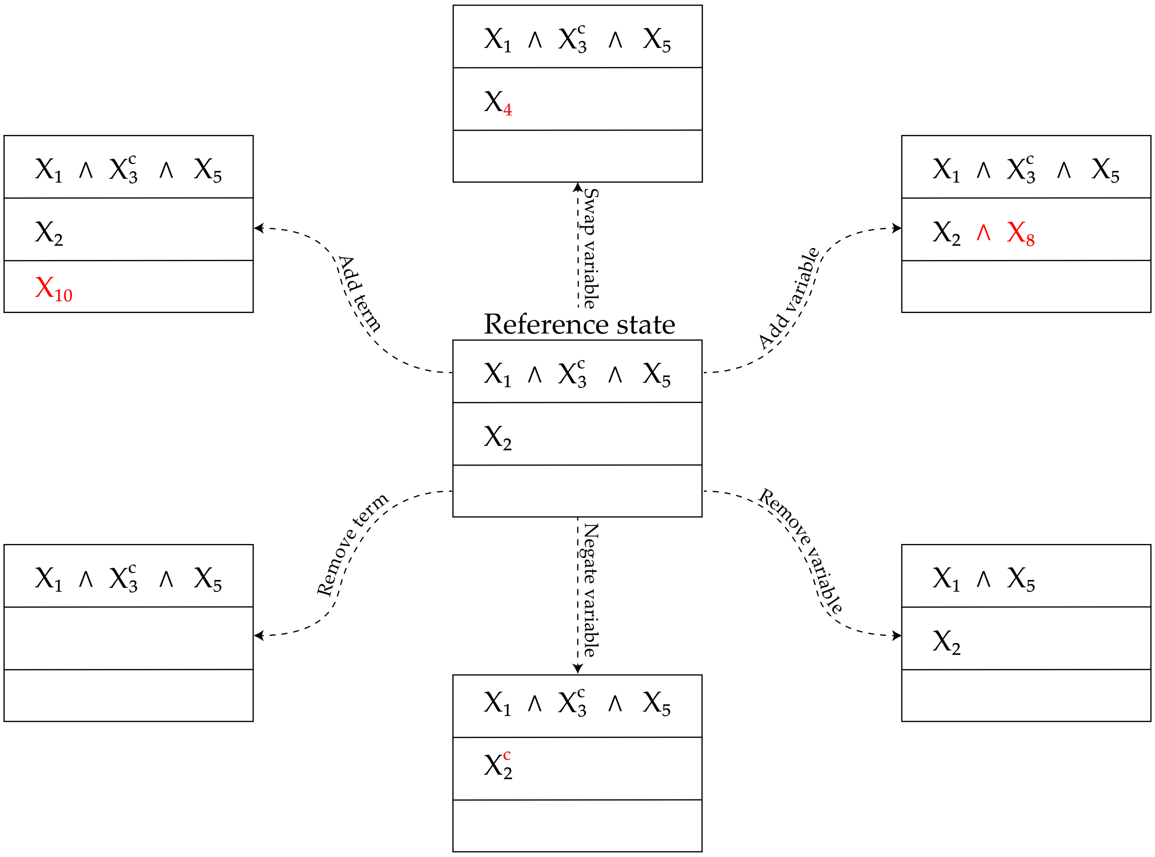

Fig. 3 Exemplary state modifications of the reference state $\{\{X_1, X_3^c, X_5\}, \{X_2\}\}$ depicted in the center

Simulated annealing is a stochastic optimization algorithm that, given a current state, randomly selects one of its neighbor states, evaluates its score, and uses the score difference between these two states for determining the probability of transitioning to the proposed neighbor state. For a state s and a proposed neighbor state s', the score function ϵ, and the current *temperature* t, this state acceptance probability is given by

$$\gamma(\epsilon(s), \epsilon(s'), t) \ := \ \min\left\{1, \exp\left(\frac{\epsilon(s) - \epsilon(s')}{t}\right)\right\}. \tag{5}$$

Thus, if a state with a better score is proposed, the transition is carried out with probability 1. However, worse states may also be accepted with the acceptance probability $\in (0, 1)$ to avoid getting stuck in local minima. The main idea of simulated annealing is slowly lowering the temperature t such that the acceptance probability of worse states tends to 0 and in the end, the globally optimal state is identified.

In logicDT, a fully automatic simulated annealing schedule governing the temperature lowering is employed. If desired, the cooling schedule can be changed, e.g., by decreasing or increasing the parameter λ, that controls the magnitude of the temperature decreases, for performing a finer or coarser stochastic search. The number of search iterations is, thus, (implicitly) controlled by λ and stopping criteria for terminating the search procedure. Alternatively, a fixed geometric cooling schedule can also be employed in logicDT. However, we recommend using the adaptive cooling schedule for fitting logicDT models. More details on the simulated-annealing-based search in logicDT are given in Appendix 1.

The proposed state modifications ensure that the modifications lead to a Markov chain that fulfills aperiodicity and irreducibility when performing a global search via simulated annealing. These properties ensure that simulated annealing asymptotically leads with probability 1 to a globally optimal state (Van Laarhoven & Aarts, 1987). More details on these Markov properties are given in Appendix 2.

3.3 The logicDT algorithm

In Algorithm 2, the logicDT procedure is presented.

```
Algorithm 2: logicDT Fitting
1  function logicDT(Training data $\mathcal{D}$):
2      $s$ = Initialize state/set of terms
3      $\mathcal{D}_s$ = Apply $s$ to $\mathcal{D}$
4      $T$ = fitDecisionTree($\mathcal{D}_s$)
5      $\text{Score}_{\text{min}} = \text{Score}(T)$
6      while Global search is not finished do
7          $s'$ = Modify current state $s$
8          $\mathcal{D}_{s'}$ = Apply $s'$ to $\mathcal{D}$
9          $T'$ = fitDecisionTree($\mathcal{D}_{s'}$)
10         $\text{Score}_{\text{new}} = \text{Score}(T')$
11         if State $s'$ is accepted based on $\text{Score}_{\text{min}}$ and $\text{Score}_{\text{new}}$ then
12             $s = s'$
13             $T = T'$
14             $\text{Score}_{\text{min}} = \text{Score}_{\text{new}}$
15         end
16     end
17     return $(s, T)$
18 end
```

In Line 2, the initial state is obtained by choosing the single input variable that minimizes the score. That is, for each input variable, a decision tree using only this input variable, i.e., a decision tree stump, is fitted and evaluated. The input variable X_j that leads to the minimum score is chosen as the initial state $\{\{X_j\}\}$. Alternatively, a random state or an empty state could also be used as the initial state.

In Lines 3 and 8, the current state is used for transforming the original training data set $\mathcal{D}$ into a tree training data set that can be directly used by a learning procedure using the identified terms as input variables. See Eq. (4) for an example on how a tree training data set is obtained from the original data set consisting of the values of the input variables.

If no leaf regression models for continuous covariables shall be fitted, the decision trees are constructed using Algorithm 1 (see Lines 4 and 9 of Algorithm 2). If leaf regression models are to be fitted (see Sect. 3.5 for more details), the splitting criterion from Sect. 3.5.2 is used in place of the impurity reduction criterion and the corresponding regression models are fitted in each leaf in contrast to single prediction values.

In Lines 5 and 10 of Algorithm 2, the training data score is calculated by passing all training observations through the fitted decision tree, performing predictions using leaf regression models if these were fitted, and comparing the predictions with the true outcomes.

In Line 7, the current state is modified by randomly performing one of the state modifications proposed in Sect. 3.2, where the state modification is randomly drawn from a uniform distribution over all possible state modifications of the current state.

This proposed modified state is then evaluated in Line 11, i.e., it is randomly accepted with the acceptance probability from Eq. (5).

The global search is carried out until a stopping criterion is true. More details on the search algorithm itself are discussed in Appendix 1.

logicDT is implemented in the R package `logicDT` (Lau, 2023) available on CRAN.

3.4 Controlling the complexity of logicDT models

For restricting the complexity of logicDT models and regularizing them, the maximum number `max_conj` of terms and the total maximum number `max_vars` of variables contained in a state should in practice be properly tuned to avoid overfitting or underfitting. Since some (potentially very long) conjunctions might correspond to no or very few observations, similar to the stopping criterion in decision trees, a *minimum conjunction size*, defining the minimum number of observations falling into this conjunction and its negation, can be specified in logicDT to exclude practically useless terms. Furthermore, one may prohibit the removal (and the addition) of whole terms in order to guarantee a certain number of terms. This might, e.g., be useful if a pure variable selection should be performed so that the maximum number of total variables is set to the maximum number of terms. In this case, the initial state should be chosen such that it already includes the desired number of terms.

logicDT aims to identify the optimal set of predictors and conjunctions with regard to the predictive ability. Thus, post-pruning of the fitted decision trees is not necessary, since the model complexity is already covered by the model size hyperparameters and the ideal splitting terms are already identified by the global search, which is similar to logic regression and in contrast to standard decision trees. However, the following two stopping criteria for locally terminating the splitting of a branch are used to filter out completely unnecessary splits.

One of the stopping criteria is the minimum number of observations in the respective leaves. If a split would lead to child nodes from which at least one of the children contains less than the prespecified number of observations, this split is prohibited. This criterion is particularly useful for regression and risk estimation purposes, where a stable estimate needs a certain amount of observations.

As second stopping criterion, the minimum (scaled) impurity reduction is considered. A split is discarded, if it does not reach the required impurity reduction, i.e., if

$$\frac{n_t}{n} \cdot \Delta i(s, t) \leq cp,$$

holds for the impurity reduction $\Delta i(s, t)$ defined in Eq. (1) and the complexity parameter $cp \geq 0$. For continuous outcomes, cp will be scaled by the empirical variance s_Y^2 of the outcome Y to ensure the right scaling, i.e., $cp \leftarrow cp \cdot s_Y^2$. Since the impurity measure for continuous outcomes is the mean squared error, this can be interpreted as controlling the minimum reduction of the normalized mean squared error (NRMSE—normalized root mean squared error—to the power of two).

The hyperparameter optimization in logicDT is discussed in more detail in Sect. 3.6.

3.5 Quantitative covariables

Decision trees are particularly suitable models for binary input data, since there is only a finite number of possible predictor scenarios in this case, i.e., every possible prediction function (including the true regression function $\mathbb{E}[Y \mid X]$) can be expressed using a decision tree. Quantitative predictors often induce a continuous relationship to the outcome that cannot be properly expressed with piecewise constant functions such as decision trees or random forests. In standard decision-tree-based methods, continuous variables are included as possible splitting candidates in the decision tree fitting process. This approach is very intuitive for merely considering all available data. However, as mentioned above, this does not allow to cover continuous relationships.

3.5.1 Leaf regression models

For properly including quantitative covariables in logicDT models, we propose, similar to MOB (model-based recursive partitioning, Zeileis et al., 2008), to fit regression models in the leaves that result from splits exclusively using the binary terms. This approach allows to fit individual curves for each binary term setting, thus, also covering interactions between the binary predictors and the quantitative covariable.

In principle, any kind of regression model such as linear or non-linear regression models could be fitted in the leaves depending on the application. Moreover, multiple regression models could also be fitted, if multiple covariables need to be considered.

For properly evaluating logicDT states, regression models need to be fitted in each decision tree and used to generate the training data predictions for computing the score, i.e., the regression models should be fitted in each iteration of the search procedure of logicDT. If, however, the computational burden is too high for, e.g., fitting non-linear regression models in each leaf of each decision tree, we recommend using linear models for the search and non-linear regression models for the final fit. In this case, the functional relationship is still taken into account in the search process and the final model utilizes the desired type of regression model. For a fast model fitting with a binary outcome, logistic regression curves through LDA (linear discriminant analysis) might be fitted that have a closed-form solution (Hastie, Tibshirani, and Friedman, 2009), and therefore, do not require an iterative optimization procedure such as standard logistic regression.

3.5.2 Splitting criterion

If regression models should be fitted in each leaf, functional trends have to be analyzed instead of simple leaf means. Therefore, we propose evaluating splits based on a likelihood-ratio test for comparing nested models as an alternative to the conventional node impurity splitting criterion specified in Eq. (1). More precisely, linear regression or LDA models, which can be determined particularly quickly, are fitted for each eligible split and resulting child node. Since we consider simple regression models, each model consists of two parameters (offset and slope) such that the difference in parameters of two submodels versus one joint model is given by $2 \cdot 2 - 2 = 2$. Thus, the likelihood-ratio test statistic

$$-2\log(\Lambda) := -2\log\left(\frac{L_{\text{reduced}}}{L_{\text{full}}}\right) \tag{6}$$

is—under the null hypothesis of equal model parameters in both subnodes—asymptotically χ^2-distributed with 2 degrees of freedom following Wilks' theorem (Wilks, 1938). Here, L_{reduced} denotes the maximized likelihood of the reduced model (i.e., the fitted joint regression model using one node) and L_{full} denotes the maximized likelihood of the full model (i.e., the model consisting of two individually fitted sub-regression models resulting in two nodes).

With the test statistic from Eq. (6), we, hence, test

$$\begin{aligned} & H_0 : \mathbb{E}[Y \mid \boldsymbol{X}_{(t)} = \boldsymbol{x}_{(t)}, X_s, E] = \mathbb{E}[Y \mid \boldsymbol{X}_{(t)} = \boldsymbol{x}_{(t)}, E] \\ \text{vs.} \quad & H_1 : \mathbb{E}[Y \mid \boldsymbol{X}_{(t)} = \boldsymbol{x}_{(t)}, X_s, E] \neq \mathbb{E}[Y \mid \boldsymbol{X}_{(t)} = \boldsymbol{x}_{(t)}, E], \end{aligned}$$

where t is the node that shall be splitted, $\boldsymbol{X}_{(t)}$ is the subvector of input variables that are used as splitting variables in ancestor nodes of t, $\boldsymbol{x}_{(t)}$ is the corresponding binary vector containing the predictor setting at node t, X_s is the binary predictor that shall be evaluated for splitting the node, and E is (are) the continuous covariable(s). We, thus, test with this likelihood-ratio test whether the split on X_s leads to different prediction models in the current tree branch. E.g., for one continuous covariable, the model

$$g(\mathbb{E}[Y \mid \boldsymbol{X}_{(t)} = \boldsymbol{x}_{(t)}, X_s, E]) = \beta_0 + \beta_1 \cdot E + \gamma_0 \cdot \mathbb{1}(X_s) + \gamma_1 \cdot \mathbb{1}(X_s) \cdot E$$

is used for testing the null hypothesis $H_0 : \gamma_0 = \gamma_1 = 0$, which is equivalent to the above null hypothesis, using the identity as link function g for a continuous outcome and the logit function as link function g for a binary outcome.

Using this new splitting criterion, likelihood-ratio tests for all eligible splits at a certain node are performed to appropriately rank eligible splits and to interpretably quantify the strength of a split. The split that achieves the lowest p-value is used, if this p-value is below a prespecified significance threshold such as $\alpha = 50\%$. Here, we propose to use a very liberal (high) threshold to avoid to miss fruitful splits. If no split can provide such a p-value, the node in question is declared as a terminal node so that this splitting criterion can also act as a stopping criterion.

Figure 4 illustrates an exemplary logicDT model with two terms and three variables in total. The current set of terms on the left induces the decision tree on the right by fitting a decision tree using the terms as potential splitting variables. The quantitative covariable E is used for evaluating the splits in likelihood-ratio tests and for fitting the regression models in the leaves. Therefore, in the root node, the terms SNP3D^c $\wedge$ SNP2D and SNP1D are both evaluated as splitting candidates by fitting regression models using E as the predictor. Since SNP3D^c $\wedge$ SNP2D yields a lower p-value than SNP1D in the likelihood-ratio test splitting criterion, the term SNP3D^c $\wedge$ SNP2D is used for splitting the root node. The fitted tree is then evaluated as a whole using a score function (see Sect. 3.2). Afterwards, the state is slightly modified using the modifications proposed in Sect. 3.2 and the procedure is repeated.

3.6 Hyperparameter optimization

For maximizing the performance of logicDT, it is necessary to optimize the model complexity parameters that act as regularization parameters. These parameters are

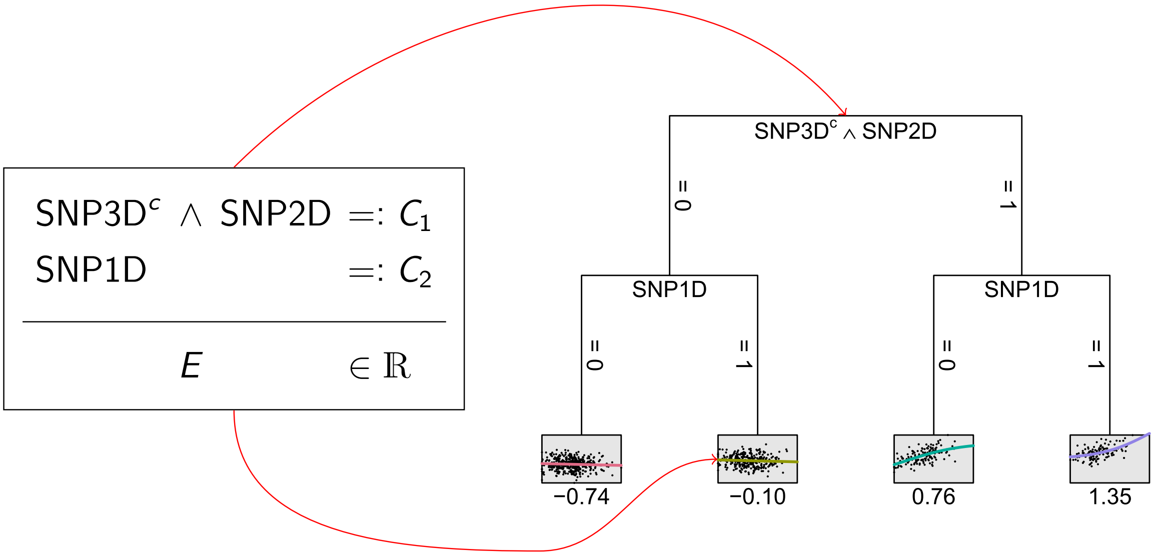

Fig. 4 An exemplary logicDT model/state. On the left hand side, the set of terms is depicted with an additional quantitative covariable which is excluded from the search over the set of terms. On the right hand side, the resulting decision trees which uses the binary predictors and identified conjunctions as input/splitting variables. Since in this case also a quantitative variable is supplied, the leaves are continuous functions instead of single point estimates

- `max_vars`—the total maximum number of variables contained in the model,
- `max_conj`—the maximum number of conjunctions/terms in the model,
- `nodesize`—the minimum number of observations per leaf in the resulting decision tree,
- `conjsize`—the minimum of observations contained in a conjunction and its negation.

In general, $\texttt{max\_vars} \geq \texttt{max\_conj}$ has to be fulfilled. Furthermore, we recommend imposing $\texttt{max\_vars} \leq 2 \cdot \texttt{max\_conj}$ in cases in which marginal effects still seem to be dominant and it is not justifiable that only high-order interaction terms compose the main influence on the outcome. This restriction is useful due to the standard learning issue that more complex models usually fit the training data better. Moreover, it reduces the set of eligible hyperparameter configurations to be evaluated speeding up the hyperparameter tuning process.

Specifically for fitting single logicDT models (via simulated annealing), it is advisable to remove the ability of removing whole conjunctions from the model in the search procedure. This ensures that the final model consists of exactly `max_conj` terms and that no extensively complex conjunctions make up the model. This also allows for a simple variable selection of marginal effects by additionally restricting $\texttt{max\_vars} = \texttt{max\_conj}$.

The purpose of `nodesize` is to ensure that each leaf contains enough observations for concluding meaningful models, i.e., stable means, or if a continuous covariable is included, regression models. A proper value for `conjsize` avoids evaluating models with uninformative conjunctions, i.e., conjunctions for which a split does not imply meaningful information due to a low number of observations. Note that for the observed values, it holds $\texttt{nodesize}_{\text{obs}} \leq \texttt{conjsize}_{\text{obs}}$, since the decision tree can further split the space. Thus, in practice, `nodesize` and `conjsize` can be set to the same value. Similar to Malley et al. (2012) who regarded probability estimation trees, we recommend a value between 1% and 10% of the total number of training observations for obtaining stable leaf estimates.

Using these parameter restrictions, a grid search evaluating all possible parameter combinations is then carried out (based on validation data) in order to identify the best setting. In Sect. 5, hyperparameter optimization following this scheme is performed.

3.7 Consistency of logicDT

In this section, we now study theoretical properties of logicDT, more precisely, the consistency of logicDT. For this purpose, we consider the core logicDT methodology, i.e., only permitting binary predictors. Without loss of generality, we assume a continuous outcome. Binary risk estimation/binary classification can be viewed as a special case using the Brier score as score function in an empirical risk minimization framework. The following theorem states that logicDT is strongly consistent. The proof of this theorem is given in Appendix 2.

Theorem 1 (Consistency of logicDT) *Suppose $\mu : \{0,1\}^p \to \mathcal{Y}$ is a p-dimensional regression function and that the outcome Y with*

$$\mathbb{E}[Y \mid \boldsymbol{X}] = \mu(\boldsymbol{X})$$

is bounded. Then, logicDT fitted via simulated annealing is strongly consistent, i.e., almost sure convergence

$$\mathbb{E}_{(\boldsymbol{X},Y)}\left[(\mu(\boldsymbol{X}) - T_n(\boldsymbol{X}))^2\right] \xrightarrow[n\to\infty]{\text{a.s.}} 0$$

holds for fitted logicDT models T_n to training data sets $\mathcal{D}_n = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$.

The following remark provides an application of Theorem 1 to hard classifications, in which the misclassification rate is evaluated.

Remark 1 For the binary classification/risk estimation case, alternatively to considering the Brier score, the excess misclassification rate is bounded by

$$\begin{aligned} 0 \ &\leq \ \mathbb{P}_{(\boldsymbol{X},Y)}(\hat{\varphi}_{T_n}(\boldsymbol{X}) \neq Y) \ - \ \mathbb{P}_{(\boldsymbol{X},Y)}(\varphi^*(\boldsymbol{X}) \neq Y) \\ &\leq \ 2\sqrt{\mathbb{E}_{(\boldsymbol{X},Y)}\left[(\mu(\boldsymbol{X}) - T_n(\boldsymbol{X}))^2\right]} \ \xrightarrow[n\to\infty]{\text{a.s.}} \ 0 \end{aligned}$$

for the classifiers $\hat{\varphi}_{T_n}(\boldsymbol{x}) = \mathbb{1}(T_n(\boldsymbol{x}) \geq 0.5)$ and the Bayes classifier φ^* (see, e.g., Theorem 1.1, Györfi et al., 2002).

Thus, the misclassification rate of the best possible classifier φ^* will be asymptotically almost surely attained by logicDT.

Note that Theorem 1 holds as long as the proposed hyperparameters are properly chosen so that the true underlying model satisfies the chosen hyperparameters. More precisely, `max_vars` and `max_conj` need to be sufficiently big and `nodesize` and `conjsize` need to be sufficiently small.

3.8 Computational complexity of logicDT

In this section, we study the computational complexity of logicDT, which is mainly controlled by the complexities of conducting a simulated-annealing-based search and fitting decision trees. A guarantee for obtaining a globally optimal model is only given if infinite iterations (or iterations in the magnitude of the size of the complete search space) are carried out in the simulated-annealing-based search (Van Laarhoven & Aarts, 1987). In practice, this is because of the size of the search space, typically, infeasible. Therefore, this asymptotic search is in practical applications approximated using a finite number of iterations (for more details on the search process, see Appendix 1). Therefore, we assume that the number of search steps is given by a finite number M.

Using the complexities of simulated annealing, decision tree fitting, and tree training data set transformation and using Algorithm 2, the computational complexity of logicDT is given in the following theorem. The proof of this theorem is given in Appendix 3.

Theorem 2 (Computational complexity of logicDT) *Suppose M is the number of search steps performed, n training observations are given, and the hyperparameters* `max_vars`, `max_conj`, `nodesize` *are fixed. Then, the computational complexity of logicDT is given by*

$$\mathcal{O}\left(Mn\left[\texttt{max\_vars} + \texttt{max\_conj}\frac{n}{\texttt{nodesize}}\right]\right).$$

Using Theorem 2, results about appropriate numbers M of search iterations based on the Markov chain length (i.e., the number of search iterations for a fixed temperature), and assumptions on the hyperparameter choices, the following corollary states that the computational complexity of logicDT is polynomial in p. The corresponding proof is, again, provided in Appendix 3.

Corollary 1 (Polynomial complexity of logicDT) *Assume that the parameters* `max_vars` *and* `max_conj` *both scale linearly with p and that the parameter* `nodesize` *is constant (with respect to n which is the worst-case scenario in which the logic decision tree may be arbitrarily deep). Further assume that the Markov chain length is fixed. Then, the computational complexity of logicDT is given by*

$$\mathcal{O}\left(n^2p^2\log(p)\right).$$

If instead the Markov chain length is chosen in the magnitude of the number of neighbor states per state (as suggested by Aarts & Van Laarhoven, 1985), the computational complexity of logicDT is given by

$$\mathcal{O}\left(n^2p^4\log(p)\right).$$

3.9 Bagged logicDT

If a single model consisting of relatively few variables cannot explain the whole variation in the outcome from the whole set of predictors or if the predictive power is of higher interest than the interpretability of the model, ensemble models consisting of several simpler models might be a preferable choice.

A particularly simple, yet effective approach is bagging (Breiman, 1996), in which for a given number of bagging iterations (e.g., 500), a single model is fitted on a random subset of the original training data set. The random subsets are typically generated via bootstrapping, i.e., performing random draws from the original training data with replacement n times. The resulting model is the ensemble of all models. Predictions are performed by averaging the predictions of the individual models. The number of iterations should, as in random forests, be chosen such that more iterations cannot reduce the generalization error substantially anymore.

Since sufficient bagging iterations are also desired in logicDT, simulated annealing with a proper amount of iterations itself might just be too slow. Moreover, the main issue of greedy search approaches, i.e., that a globally optimal state could be missed due to being stuck in a local optimum, might be diminished through considering different subsets of the training data set and stabilizing the model over them. In other words, the variance stabilizing property of bagging might be sufficient to account for the drawbacks of a greedy search (Murthy & Salzberg, 1995).

For the usage of logicDT in an ensemble framework, we, therefore, propose a greedy search for fitting individual logicDT models. In this greedy search, the same state modifications as in the simulated-annealing-based search are used (see Sect. 3.2). In contrast to simulated annealing, the greedy search deterministically chooses the best neighbor in each iteration. Thus, for each current state, all its neighbors are evaluated and the neighbor with the lowest score amongst all neighbors is chosen as new state. Note that for increasing numbers of predictors and increasing numbers of allowed terms and total variables, the number of eligible neighbors per state increases quadratically, thus, slowing down the greedy search. For handling higher-dimensional data, a randomization of the greedy search might be a solution which we, however, did not consider in this article.

Another very useful property of bagging is that in the fitting of an individual model not all observations from the training data are employed. The not considered observations called oob (out-of-bag) observations can, therefore, be used to estimate the generalization error, similar to using independent test data. This estimate is called the oob error and is obtained by only using models that were not built using the considered observation. More precisely, the oob error is calculated by averaging over the oob errors of the observations, where the oob error of an observation can be computed by only choosing the models which did not use this observation for training and by temporarily constructing an ensemble from this subset of models for predicting the outcome of this observation. In particular, for the estimation of variable importance measures (VIMs), bagging and oob observations are very beneficial. As discussed in the following section, we, therefore, also use them in the construction of the VIM considered in logicDT.

4 Variable importance measures

In many applications, it is useful to measure the influence of the input variables or their interactions on the prediction of an outcome. Variable importance measures (VIMs) directly try to quantify this influence. Typically, this influence is estimated by comparing two models, namely

- the original full model containing the term of interest and

- a kind of informatively reduced model, in which the term of interest no longer plays an informative role.

Then, the difference between the prediction errors of these two models is computed and is taken as an estimate of how the prediction based on the model improves if the term is properly included, where the prediction errors are, e.g., given by the mean squared error in regression tasks or $1 - \mathrm{AUC}$ in binary risk estimation tasks.

4.1 Computation of VIMs

Let $\epsilon(\tilde{\boldsymbol{X}})$ be a prediction error measure capturing the performance of a fitted model informatively using only the input variables in $\tilde{\boldsymbol{X}} \subseteq \boldsymbol{X}$, interpreting the random vector of input variables $\boldsymbol{X} = (X_1, \dots, X_p)$ as a set $\boldsymbol{X} = \{X_1, \dots, X_p\}$. Then, the importance of an input variable X_i is given by

$$\mathrm{VIM}(X_i) = \epsilon(\boldsymbol{X} \setminus X_i) - \epsilon(\boldsymbol{X}). \tag{7}$$

Here, $\epsilon(\boldsymbol{X} \setminus X_i)$ describes the prediction error of the reduced model informatively excluding the variable X_i and $\epsilon(\boldsymbol{X})$ describes the prediction error of the original full model.

Bagging allows the unbiased estimation of VIMs on the full training data set by performing oob predictions. Moreover, bagging also has the advantage that multiple potentially different models are explored stabilizing the VIMs themselves. Thus, for estimating VIMs in logicDT, bagging is used and the discussed VIMs are computed on the oob observations.

4.1.1 Permutation VIM and removal VIM

One particularly popular approach for estimating the reduced model is the *permutation VIM* used in random forests (Breiman, 2001). In this approach, for estimating the importance of a certain input variable, its corresponding observations are randomly permuted and predictions with this random permutation are performed. Typically, the VIM data set is permuted multiple times in the specific predictor and the average prediction error of these permutations is compared against the original error.

As an alternative, the reduced model can also be directly fitted using a reduced training data set from which the predictor of interest was removed (Mentch & Hooker, 2016). In the following, we call this approach the *removal VIM*.

4.1.2 Logic VIM

For binary predictors, we additionally propose a specific third procedure for computing VIMs. The idea of this *logic VIM* is based on considering each possible predictor setting of the input variable of interest equally, i.e., for a binary predictor $X_1 \in \{0, 1\}$, the error of the reduced model is estimated by performing predictions fixing $X_1 = 0$, performing predictions fixing $X_1 = 1$ and averaging these predictions before computing the error. Thus, for each observation, the prediction of the reduced model considers both possible decision tree paths, one for $X_1 = 0$ and one for $X_1 = 1$, equally and is generated without knowledge about X_1.

4.2 Adjustment for interactions

In standard VIM procedures such as the permutation VIM in random forests, only importances of single input variables are considered. In the context of logicDT, we measure the importance of terms, i.e., of identified single input variables or conjunctions of several input variables. For instance, if the resulting model consists of $\{\{X_1\}, \{X_2 \wedge X_3^c\}\}$, we are interested in specifying the importance of X_1 as well as the importance of the term $X_2 \wedge X_3^c$. This is achieved by considering terms such as $X_2 \wedge X_3^c$ as single input variables, i.e., by directly considering a tree training data set as in Eq. (4).

Since decision trees can handle interactions themselves, it might be possible that, e.g., X_1 as well as the interaction $X_1 \wedge X_2^c$ exhibit strong effects on the outcome. However, due to the strong marginal effect, only the single predictors X_1 and X_2 might be included in the logicDT model, complicating the estimation of the importance of the interaction.

Hence, we propose a novel VIM adjustment procedure for interactions that quantifies the importance of interactions that were not identified by a supervised learner such as logicDT. This VIM adjustment approach presented in the following does not depend on logicDT, but enables logicDT to appropriately estimate interaction importances. Therefore, they could, in principle, be applied to all black-box models for estimating interaction importances.

The idea behind the VIM adjustment procedure is based on considering several predictors at once, i.e., the reduced model results from reducing multiple variables in one step. Comparing the performances of this reduced model and the original model yields a joint VIM of the set of predictors (Bureau et al., 2005). Analogously to Eq. (7), the joint VIM is obtained by

$$\mathrm{VIM}(X_{i_1}, \dots, X_{i_k}) \;=\; \epsilon(\boldsymbol{X} \setminus \{X_{i_1}, \dots, X_{i_k}\}) - \epsilon(\boldsymbol{X}). \tag{8}$$

Since this joint VIM still includes the marginal effects of the individual predictors and their sub-interactions of an order lower than the order of the actual interaction influencing the outcome, we propose the *interaction VIM* that corrects for any effects contained in the regarded interaction. This interaction VIM of $X_{i_1} \wedge \cdots \wedge X_{i_k}$ is given by

$$\begin{aligned}\mathrm{VIM}(X_{i_1} \wedge \cdots \wedge X_{i_k}) &= \mathrm{VIM}(X_{i_1}, \dots, X_{i_k} \mid \boldsymbol{X} \setminus \boldsymbol{Z}) \\ &\quad - \sum_{\{j_1,\dots,j_l\} \subsetneq \{i_1,\dots,i_k\}} \mathrm{VIM}(X_{j_1} \wedge \cdots \wedge X_{j_l} \mid \boldsymbol{X} \setminus \boldsymbol{Z}),\end{aligned} \tag{9}$$

where $\boldsymbol{Z} := \{X_{i_1}, \dots, X_{i_k}\}$ is the set of input variables in the considered interaction. In our notation, $\wedge$ denotes the interaction importance, while commas represent the joint importance. By $\mathrm{VIM}(\boldsymbol{A} \mid \boldsymbol{X} \setminus \boldsymbol{Z})$, the VIM of $\boldsymbol{A}$ considering the predictor set excluding the variables in $\boldsymbol{Z}$, i.e., the improvement of additionally considering $\boldsymbol{A}$, while regarding only the predictors in $\boldsymbol{X} \setminus \boldsymbol{Z}$, is denoted. The interaction importance captures the importance of a general meaning of interaction, i.e., it considers whether some variables do interact in any way and quantifies the effect of the joint presence of these variables adjusted for single occurrences. For a predictor set $\tilde{\boldsymbol{A}} := \{X_{j_1}, \dots, X_{j_l}\} \subseteq \boldsymbol{Z}$, the restricted joint VIM, i.e., the VIM of $\tilde{\boldsymbol{A}}$ considering only the predictors $\boldsymbol{X} \setminus \boldsymbol{Z}$ in the reduced model, is, following Eq. (8), given by

$$\mathrm{VIM}(\tilde{\boldsymbol{A}} \mid \boldsymbol{X} \setminus \boldsymbol{Z}) \;=\; \epsilon(\boldsymbol{X} \setminus \boldsymbol{Z}) - \epsilon(\tilde{\boldsymbol{A}} \cup (\boldsymbol{X} \setminus \boldsymbol{Z})). \tag{10}$$

Excluding all variables in $\boldsymbol{Z}$ composing the interaction in the respective reference models is crucial for isolating the effects that should be adjusted for. If, e.g., a two-way interaction $X_1 \wedge X_2$ is studied, its interaction VIM (9) is given by

$$\begin{aligned}\mathrm{VIM}(X_1 \wedge X_2) &= \mathrm{VIM}(X_1, X_2 \mid \boldsymbol{X} \setminus \{X_1, X_2\}) \\ &\quad - \mathrm{VIM}(X_1 \mid \boldsymbol{X} \setminus \{X_1, X_2\}) - \mathrm{VIM}(X_2 \mid \boldsymbol{X} \setminus \{X_1, X_2\}).\end{aligned} \tag{11}$$

If, e.g., $\mathrm{VIM}(X_1 \mid \boldsymbol{X} \setminus X_1) \overset{(7),(10)}{=} \mathrm{VIM}(X_1)$ would be used instead of $\mathrm{VIM}(X_1 \mid \boldsymbol{X} \setminus \{X_1, X_2\})$ in Eq. (11), the whole importance of X_1, that also contains the interaction with X_2, would be subtracted from the joint importance not isolating the interaction importance that should be estimated.

Recursively applying Eq. (11) to the general case in Eq. (9) yields

$$\mathrm{VIM}(X_{i_1} \wedge \cdots \wedge X_{i_k}) = \sum_{\{j_1,\ldots,j_l\} \subseteq \{i_1,\ldots,i_k\}} (-1)^{k-l} \cdot \mathrm{VIM}(X_{j_1}, \ldots, X_{j_l} \mid \boldsymbol{X} \setminus \boldsymbol{Z}).$$

Utilizing Eq. (10), this formula for the interaction VIM can also be written in terms of prediction errors ϵ, i.e., as

$$\mathrm{VIM}(X_{i_1} \wedge \cdots \wedge X_{i_k}) = \sum_{\{j_1,\ldots,j_l\} \subseteq \{i_1,\ldots,i_k\}} (-1)^{l+1} \cdot \epsilon(\boldsymbol{X} \setminus \{X_{j_1}, \ldots, X_{j_l}\}) - \epsilon(\boldsymbol{X}).$$

This formula can be used for efficiently computing the interaction VIM by directly considering prediction errors.

The interaction VIM (9) is similar to the interaction effect statistic proposed by Friedman and Popescu (2008), which utilizes the same effect decomposition and is based on the explained variance of partial dependence functions instead of VIMs. Friedman and Popescu (2008) theoretically justified this effect decomposition by showing that their statistic is zero, if the null hypothesis of no present interaction effect holds true. For example, for analyzing a two-way interaction $X_1 \wedge X_2$, Friedman and Popescu (2008) evaluate $F_{X_1,X_2} - F_{X_1} - F_{X_2}$, in which $F_{\cdot}$ denotes partial dependence functions of the considered input variables. This term is analogous to the interaction VIM in Eq. (11) for $X_1 \wedge X_2$ with the difference that VIMs, i.e., performance metrics, are used instead of partial dependence functions. Moreover, the input feature effect decomposition utilized by the proposed interaction VIM is also used by the Shapley interaction index (Lundberg et al., 2020; Fujimoto et al., 2006). However, in machine learning applications, Shapley values are based on direct predictions of the fitted model instead of performance metrics such as VIMs.

For all three procedures for constructing VIMs mentioned in Sect. 4.1, the reduced joint model can be intuitively constructed.

In the permutation VIM, the input variables of interest, i.e., the input variables participating in the interaction for which the interaction VIM should be computed, are simply permuted together by, e.g., permuting the values of each input variable separately.

For the removal VIM, the set of input variables of interest is removed as a whole from the total set of input variables.

The logic VIM proposed in Sect. 4.1.2 performs uninformative predictions of an input variable by considering both possible decision tree paths for an observation and averaging the prediction. To generalize the logic VIM to multiple input variables at once for computing the interaction VIM, all possible predictor settings $\boldsymbol{x} \in \{0, 1\}^p$ for the p input variables that shall be informatively excluded are used to generate predictions. These 2^p predictions are averaged to create the prediction of the reduced model.

In logicDT, the logic VIM is used in conjunction with the proposed adjustment for interaction effects. Quantifying the importance of specific conjunctions, that are, e.g., identified by logicDT, will be discussed in the following section. In Sect. 5, the permutation VIM, the removal VIM, and the logic VIM are evaluated in empirical studies.

4.3 Adjustment for conjunctions

The VIM adjustment approach introduced in Sect. 4.2 only captures the importance of a general meaning of interactions, i.e., it just considers the question whether some variables do interact in some way. Since logicDT is aimed at identifying specific conjunctions (and also determines the values of a VIM for them, if the conjunctions have been identified by logicDT), a further adjustment approach is proposed that tries to identify the specific conjunction leading to an interaction effect. For example, if the importance of the interaction between X_1 and X_2 was quantified using the interaction adjustment proposed in Sect. 4.2, the approach presented in the following assigns a Boolean conjunction to this importance, e.g., the Boolean conjunction $X_1 \wedge X_2^c$. The proposed procedure is, again, applicable to any kind of supervised learning model. However, due to considering Boolean conjunctions, the input variables for which the importance should be quantified need to be binary.

This approach considers each possible conjunction of the identified interaction and chooses the conjunction that leads to the most severe deviation in the outcome, i.e., the conjunction with the strongest effect on the outcome. The VIM of this conjunction is the corresponding interaction VIM derived in Sect. 4.2.

The idea of this method is to consider the values of the outcome for each possible scenario of the interacting variables, e.g., for $X_1 \wedge (X_2^c \wedge X_3)$, where we assume that the terms X_1 and $X_2^c \wedge X_3$ were identified by logicDT. In this example, thus, two interacting terms are regarded, i.e., the $2^2 = 4$ possible scenarios $X_1 = 0$ or $X_1 = 1$ in combination with $X_2^c \wedge X_3 = 0$ or $X_2^c \wedge X_3 = 1$ are considered. For each setting, the corresponding outcome values are compared to the outcome values of the complementary set, i.e., the set in which the considered conjunction is equal to zero. This means that in the considered example the four statistical tests

$$\begin{aligned} & H_0 : \mathbb{E}\big[Y \mid C_i = 1\big] = \mathbb{E}\big[Y \mid C_i = 0\big] \\ \text{vs.} \quad & H_1 : \mathbb{E}\big[Y \mid C_i = 1\big] \neq \mathbb{E}\big[Y \mid C_i = 0\big], \end{aligned}$$

with

$$\begin{aligned} C_1 &= X_1 \wedge (X_2^c \wedge X_3), \quad & C_2 &= X_1^c \wedge (X_2^c \wedge X_3), \\ C_3 &= X_1 \wedge (X_2^c \wedge X_3)^c, \quad & C_4 &= X_1^c \wedge (X_2^c \wedge X_3)^c \end{aligned}$$

potentially negating the subterms, are performed for $i \in \{1, 2, 3, 4\}$. For continuous outcomes, Welch's t-test is performed for comparing the means between these two groups, i.e., the group in which the considered conjunction is equal to one and the group in which the considered conjunction is equal to zero. For binary outcomes, Fisher's exact test is performed for testing different underlying case probabilities. The combination with the lowest p-value is chosen as the explanatory term for the interaction effect. E.g., in the above example, if the smallest p-value results from considering $X_1 = 0$ and $(X_2^c \wedge X_3) = 1$, the term $X_1^c \wedge (X_2^c \wedge X_3)$ is chosen as the conjunction responsible for the interaction effect.

5 Experiments

In the following, we evaluate the performance of logicDT on simulated and real data considering classification and regression problems and compare logicDT with other similar methods. More precisely, we compare logicDT and bagged logicDT with conventional decision trees (Breiman et al., 1984), DL8.5 (Aglin et al., 2020a), random forests (Breiman, 2001), gradient boosting (Friedman, 2001), logic regression (Ruczinski et al., 2003), logic regression with bagging (Schwender & Ickstadt, 2007), MOB (model-based recursive partitioning, Zeileis et al., 2008), interaction forests (Hornung & Boulesteix, 2022), and RuleFit (Friedman & Popescu, 2008). Since DL8.5 (as similar openly available optimal decision tree algorithms such as MurTree proposed by Demirović et al., 2022) are currently only implemented for classification tasks, DL8.5 is only applied to the considered classification tasks. All analyses are carried out using `R` (R Core Team, 2022), except for the application of DL8.5, which is performed using the `Python` implementation of Aglin et al. (2020b).

5.1 Simulation study

We, first, consider the situation of genetic association studies in which single genes/genetic pathways are analyzed and typically not more than a few tens of SNPs (single nucleotide polymorphisms) are considered. Afterwards, we consider a more complex setting with more SNPs to evaluate if logicDT is also applicable to high-dimensional problems.

5.1.1 First simulation setup

We analyze the performance of logicDT and the other supervised learning procedures first in four different simulation scenarios in which we consider binary predictors and

- a binary outcome (such as a disease status) without an additional continuous covariable,
- a binary outcome with a continuous covariable,
- a continuous outcome (such as the blood pressure) without a continuous covariable, and
- a continuous outcome with a continuous covariable.

Our simulations are based on the problem of analyzing risk factors in genetic epidemiology. Thus, the generated input variables can be interpreted as SNPs that count the number of minor alleles at a specific locus, i.e., the number of occurrences of a less frequent base-pair substitution at a specific location in the DNA. Due to humans being diploid organisms, i.e., carrying two complete sets of chromosomes, SNPs can take the values 0, 1, or 2. Similar to, e.g., logic regression, for the application of logicDT to SNP data, each SNP is divided into the binary input variables $\text{SNP}_D = \mathbb{1}(\text{SNP} \neq 0)$ and $\text{SNP}_R = \mathbb{1}(\text{SNP} = 2)$, coding for a dominant and a recessive effect, respectively, such that no information is lost. Conventional decision trees also implicitly divide SNPs into dominant and recessive effects by considering SNPs as numerical variables such that a split can occur on $(\{0\}, \{1, 2\})$ or on $(\{0, 1\}, \{2\})$. Combined with the greedy search of decision trees over all possible splits, this is equivalent to directly considering the binary variables SNP_D and SNP_R (Lau et al., 2022).

The genotypes of the SNPs are generated independently, resembling sets of SNPs from which, as often done in practice, highly correlated SNPs have been removed using linkage-disequilibrium-based pruning (see, e.g., Purcell et al., 2007). The distributions of the SNPs are defined via the MAF (minor allele frequency), i.e., the proportion of minor allele occurrences, yielding the binomial distribution $\text{Bin}(2, \text{MAF})$ for each SNP. For all simulated SNPs, we consider a MAF of 0.25. For each data set, 50 SNPs are generated so that $\boldsymbol{X} = (\mathrm{SNP}_1, \dots, \mathrm{SNP}_{50})$. However, in the considered scenarios described below, only a small fraction influences the outcome such that most input variables act as noise regarding the outcome.

For the analysis of the influence of a continuous covariable, an environmental variable (e.g., an air pollution indicator) is generated following a truncated normal distribution (truncated at zero, since values below zero often do not occur in practice). In particular, the environmental term E is generated by considering a $\mathcal{N}(20, 100)$-distributed random variable E' and setting values below zero to zero so that $E = \max(0, E')$. The truncated values might, e.g., be interpreted as measurements below a detection limit.

Since DL8.5 can only incorporate binary input variables, E is dichotomized into a binary variable by considering $E_{\mathrm{bin}} = \mathbb{1}(E > 20)$ for fitting and evaluating DL8.5 models, where the cutoff 20 is chosen due to $\mathbb{E}[E] = \mathbb{P}(E' > 0)\mathbb{E}[E' \mid E' > 0] \approx 20$.

For the first simulation scenario considering a binary outcome without any continuous covariables, the outcome is generated following the model

$$\begin{aligned}\mathrm{logit}(\mathbb{P}(Y = 1 \mid \boldsymbol{X})) \; = \; -0.4 + \Big(&\sqrt{\log(1.5)} \cdot \mathbb{1}(\mathrm{SNP}_1 > 0) \\ &+ \sqrt{\log(2)} \cdot \mathbb{1}(\mathrm{SNP}_2 > 0 \land \mathrm{SNP}_3 = 0)\Big)^2.\end{aligned}$$

Therefore, SNP_1 exhibits a moderate marginal effect and SNP_2 and SNP_3 interact with each other. The linear predictor on the right-hand side is squared which means that, on the scale of the total linear predictor, the term $\mathbb{1}(\mathrm{SNP}_1 > 0)$ interacts with the term $\mathbb{1}(\mathrm{SNP}_2 > 0 \land \mathrm{SNP}_3 = 0)$. Thus, this resembles a situation in which it might be useful to be able to model interactions between interactions, since the underlying scale of the linear predictor is unknown prior to the analyses, which is usually the case in practice. The intercept of −0.4 ensures that the resulting data sets are approximately balanced, i.e., that the fraction of cases is approximately equal to 50%.

In the second scenario, a gene-environment interaction is introduced such that the outcome in this case is modeled by

$$\begin{aligned}\mathrm{logit}(\mathbb{P}(Y = 1 \mid \boldsymbol{X}, E)) \; = \; &-0.45 + \log(2) \cdot \mathbb{1}(\mathrm{SNP}_1 > 0) \\ &+ \log(3) \cdot \frac{E}{20} \cdot \mathbb{1}(\mathrm{SNP}_2 > 0 \land \mathrm{SNP}_3 = 0).\end{aligned}$$

Thus, the environmental variable only influences the outcome, if the term $\mathbb{1}(\mathrm{SNP}_2 > 0 \land \mathrm{SNP}_3 = 0)$ holds true. This kind of gene-environment interaction might be reasonable for substances that are usually harmless, but might cause, e.g., allergic reactions in individuals with a certain genetic makeup.

Analogously to the first scenario, the third scenario consists of data sets in which the outcome is modeled by

$$\begin{aligned}\mathbb{E}[Y \mid \boldsymbol{X}] \; = \; -0.4 + \Big(&\sqrt{\log(1.5)} \cdot \mathbb{1}(\mathrm{SNP}_1 > 0) \\ &+ \sqrt{\log(2)} \cdot \mathbb{1}(\mathrm{SNP}_2 > 0 \land \mathrm{SNP}_3 = 0)\Big)^2.\end{aligned}$$

Here and in the following scenario, random noise generated from $\mathcal{N}(0, 1)$ is added to the linear predictor.

As in the second scenario, the fourth scenario follows the underlying model

$$\begin{aligned}\mathbb{E}[Y \mid \boldsymbol{X}, E] \;=\;& -0.75 + \log(2) \cdot \mathbb{1}(\mathrm{SNP}_1 > 0) \\ & + \log(4) \cdot \frac{E}{20} \cdot \mathbb{1}(\mathrm{SNP}_2 > 0 \wedge \mathrm{SNP}_3 = 0).\end{aligned}$$

For each simulation scenario, 100 independent data sets are generated. For each data set, it is assumed that this is the only data set available. Thus, for each replication, the data set is randomly divided into a training, a validation, and a test data set. Thus, for the evaluation of logicDT and comparable methods, we perform 100 independent evaluations. In many practical applications such as in the construction of genetic risk scores, there is only data for a relatively small number of observations available. Hence, in our simulations, the randomly generated data sets consist of 1000 observations each. From each of these data sets, $0.7 \cdot 1000 = 700$ randomly chosen observations are used as the intermediary data set for training and validating the model and the remaining 300 observations yield the test data set for the final evaluation. The intermediary data set is further randomly divided into $0.25 \cdot 700 = 175$ observations for choosing the best set of hyperparameters and $0.75 \cdot 700 = 525$ observations for training in the hyperparameter optimization. After the optimal hyperparameter setting has been identified, the final models are trained on the intermediary data set consisting of 700 observations.

The predictive performance of logicDT and the comparable methods is assessed using the AUC for binary outcomes and using the complement of the NRMSE (normalized root mean squared error) for continuous outcomes on test data predictions.

5.1.2 Hyperparameter optimization

As described in Sect. 3.6, the model complexity parameters `max_vars` (maximum number of total variables) and `max_conj` (maximum number of conjunctions) of logicDT should be tuned. In this application, we prohibit removing complete conjunctions to ensure that the models consist of exactly `max_conj` conjunctions. Furthermore, the minimum number `nodesize` of observations belonging to a leaf and the minimum number `conjsize` of observations belonging to a conjunction and its negation are tuned using the same value, respectively. This ensures that the trees are grown to the ideal depth and prevents that models using uninformative conjunctions are evaluated.

For bagged logicDT models, `max_vars` and `max_conj` are tuned using the same parameter setting and allowing the removal of complete conjunctions in contrast to fitting single logicDT models.

In Table 1, the considered hyperparameter settings for logicDT, bagged logicDT, and the comparable tree-based statistical learning methods are summarized. For logicDT, the hyperparameter settings proposed in Sect. 3.6 are considered. For the regarded comparable methods, common hyperparameter choices are considered and the best performing one is chosen. For all methods except for gradient boosting and RuleFit, a grid search among all proposed settings is performed, due to relatively few plausible settings. For gradient boosting and RuleFit, a sequential Bayesian hyperparameter search is carried out (Bergstra et al., 2011; Wilson, 2021), since a finetuning of the learning rate parameter (for a fixed number of boosting iterations) is required. Additionally, the subsample fraction and the minimum node size, which can also be considered as continuous hyperparameters, have to be configured jointly in gradient boosting and RuleFit. For this sequential search, 100 different settings are evaluated.

Table 1 Regarded hyperparameter settings with according descriptions

Algorithm	Software package	Hyperparameter	Description	Considered realizations
logicDT	`logicDT` (Lau, 2023)	(`max_vars`, `max_conj`)	Maximum number of variables and maximum number of conjunctions	$\{(i,j) \in \{1,\dots,10\}^2 \mid i \geq j \wedge i \leq 2j\}$
		`nodesize`/`conjsize`	Minimum number of observations per leaf and minimum number of observations per conjunction	$\lfloor\{0.01, 0.05, 0.1\} \cdot N\rfloor$
logicDT–Bagging	`logicDT` (Lau, 2023)	(`max_vars`, `max_conj`)	Maximum number of variables and maximum number of conjunctions	$\{(1,1),\dots,(10,10)\}$
		`nodesize`/`conjsize`	Minimum number of observations per leaf and minimum number of observations per conjunction	$\lfloor\{0.01, 0.05, 0.1\} \cdot N\rfloor$
		`bagging.iter`	Number of bagging iterations	500
Decision Tree	`rpart` (Therneau & Atkinson, 2019)	`cp`	Complexity parameter for optimal pruning	$\{0, 10^{-3}, 10^{-2.5}, 10^{-2}, 10^{-1.5}, 10^{-1}, 10^{-0.5}, 10^{0}\}$
		`minbucket`	Minimum number of observations per leaf	$\lfloor\{0.01, 0.05, 0.1\} \cdot N\rfloor$
DL8.5	`PyDL8.5` (Aglin et al., 2020b)	`min_sup`	Minimum number of observations per leaf	$\lfloor\{0.01, 0.05, 0.1, 0.2\} \cdot N\rfloor$
Random Forests	`ranger` (Wright & Ziegler, 2017)	`mtry`	Number of randomly drawn input variables at each split	$\left\lfloor\{0.5, 1, 2\} \cdot \lfloor\sqrt{p}\rfloor\right\rfloor \cup \left\lfloor\{0.5, 1, 2\} \cdot \lfloor\frac{p}{3}\rfloor\right\rfloor$
		`min.node.size`	Minimum number of observations per leaf	$\lfloor\{0.01, 0.05, 0.1\} \cdot N\rfloor$
		`num.trees`	Number of trees grown	2000
Gradient Boosting	`xgboost` (Chen & Guestrin, 2016)	`subsample`	Subsample fraction for drawing samples without replacement for each tree	[0.5, 1.0]
		`min_child_weight`	Minimum number of observations per leaf	$\{m \in \mathbb{N} \mid m \in [0.01 \cdot N, 0.1 \cdot N]\}$
		`eta`	Learning rate	$[10^{-6}, 10^{-1}]$
		`nrounds`	Number of boosting iterations	500

Table 1 (continued)

Algorithm	Software package	Hyperparameter	Description	Considered realizations
Logic Regression	`LogicReg` (Kooperberg & Ruczinski, 2022)	`(nleaves, ntrees)`	Maximum number of (total) leaves and maximum number of trees	$\{(i,j) \in \{1,\ldots,20\} \times \{1,\ldots,5\} \mid i \geq j\}$
		`anneal.control`	Simulated annealing cooling schedule	Experimental
Logic Bagging	`logicFS` (Schwender & Ickstadt, 2007)	`(nleaves, ntrees)`	Maximum number of (total) leaves and maximum number of trees	$\{(i,j) \in \{1,\ldots,10\} \times \{1,\ldots,5\} \mid i \geq j\}$
		`B`	Bagging iterations	500
MOB	`party` (Zeileis et al., 2008)	`minsplit`	Minimum number of observations per leaf	$\lfloor \{0.01, 0.02, \ldots, 0.09, 0.1\} \cdot N \rfloor$
Interaction Forests	`diversityForest` (Hornung, 2022)	`npairs`	Number of randomly drawn input variable pairs at each split	$\left\lfloor \{0.5, 1, 2\} \cdot \lfloor \sqrt{p} \rfloor \right\rfloor \cup \left\lfloor \{0.5, 1, 2\} \cdot \lfloor \frac{p}{3} \rfloor \right\rfloor$
		`min.node.size`	Minimum number of observations per leaf	$\lfloor \{0.01, 0.05, 0.1\} \cdot N \rfloor$
		`num.trees`	Number of trees grown	2000
RuleFit	`pre` (Fokkema, 2020)	`sampfrac`	Subsample fraction for drawing samples without replacement for each tree	[0.5, 1.0]
		`minbucket`	Minimum number of observations per leaf	$\{m \in \mathbb{N} \mid m \in [0.01 \cdot N, 0.1 \cdot N]\}$
		`learnrate`	Learning rate	$[10^{-6}, 10^{-1}]$
		`ntrees`	Number of boosting iterations	500

The mentioned hyperparameter names are the names of the corresponding arguments in the respective `R`/`Python` packages

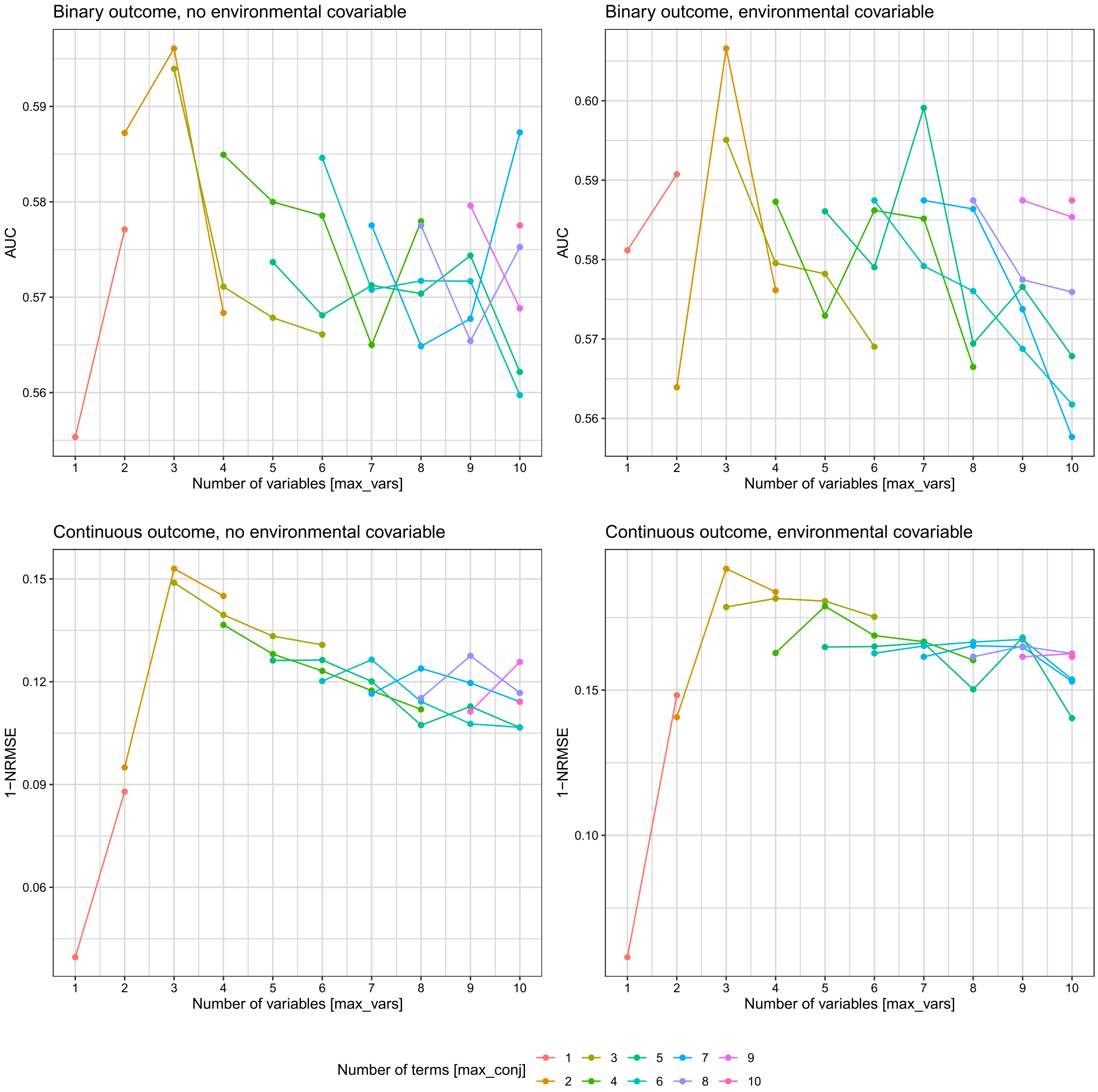

Fig. 5 Predictive performances of different hyperparameter settings for the parameters `max_vars` (maximum number of variables) and `max_conj` (maximum or exact number of terms) in logicDT in the simulation study considering four different scenarios. The performance for binary outcomes is measured by the AUC and the performance for continuous outcomes is measured by the complement of the NRMSE (normalized root mean squared error). Results on validation data sets for the best respective setting of the parameter `nodesize`/`conjsize` in the set $\{1\%, 5\%, 10\%\}$. The evaluated hyperparameter settings are listed in Table 1. Justifications for evaluating these settings are given in Sect. 3.6

For logicDT, Fig. 5 shows the validation data performances for the considered settings of `max_vars` and `max_conj` combined with the respective best setting for `nodesize` /`conjsize`. For each scenario, the highest performance is yielded by `max_vars` = 3 and `max_conj` = 2 corresponding to the true underlying simulation models. Generally, the following pattern can be observed. For many `max_conj` settings, the maximizing setting is given by `max_vars` = `max_conj` + 1, which is due to the fact that in this case, additionally to single variables as terms, a conjunction of two variables is contained in the model.

For most considered hyperparameter settings, the validation performance does not seem to vary too severely between similar settings, which indicates that a slight hyperparameter misspecification might not substantially impair the predictive performance of the resulting logicDT model.

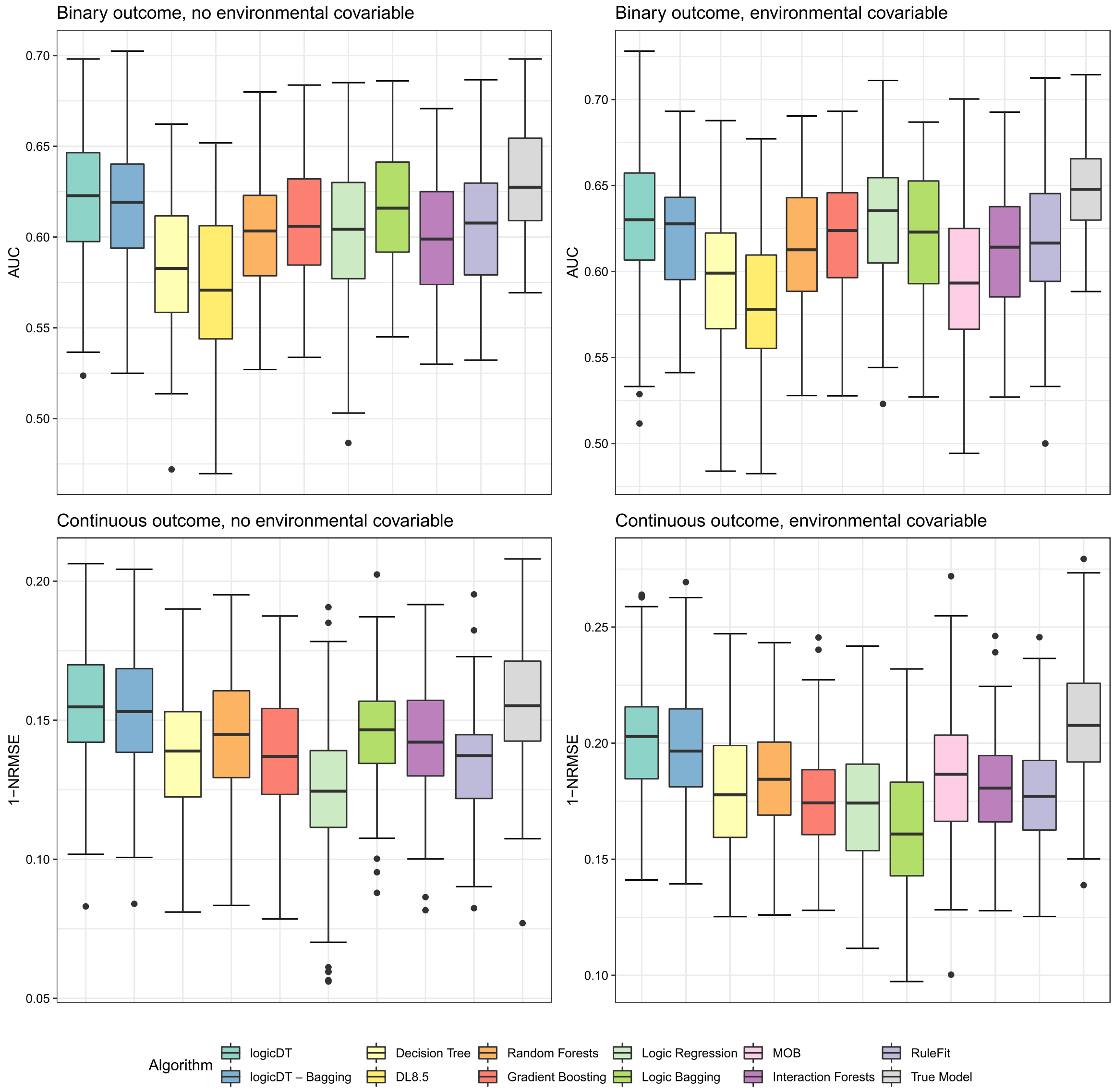

Fig. 6 Predictive performances of logicDT and the comparable methods in the simulation study considering four different scenarios. The performance for binary outcomes is measured by the AUC and the performance for continuous outcomes is measured by the complement of the NRMSE (normalized root mean squared error)

5.1.3 Predictive performance

Figure 6 depicts the performances of logicDT, the comparable methods, and the true underlying model in the simulation study, where the performance of the true model was assessed by performing predictions using the true regression functions presented in Sect. 5.1.1.

In the first simulation scenario considering a binary outcome without an environmental covariable, most notably, standard logicDT and bagged logicDT lead to the best performances, i.e., the largest AUC values, which almost coincide with the performance of the true model. Among the comparable methods, logic bagging seems to be the best method.

For the second scenario in which also a gene-environment interaction is considered, logicDT, bagged logicDT, gradient boosting, logic regression, and logic bagging induce similar results superior to the remaining methods. Here, logicDT and logic regression seem to produce slightly better results than the other procedures.

For the third and fourth simulation scenarios considering a continuous outcome without or with an environmental covariable, logicDT and bagged logicDT yield the highest predictive performances close to the true underlying models. When considering no environmental covariable, logic bagging seems to be the best method among the comparable methods. MOB yields the highest performance among the comparable methods when including an environmental covariable.

5.1.4 Variable importance

Using the VIMs and adjustment approaches for interactions and conjunctions proposed in Sect. 4, we computed variable importances in the four different simulation scenarios. We fitted bagged logicDT models on the 100 complete sub data sets for each scenario. The VIMs themselves were computed using out-of-bag data. For properly summarizing the 100 repetitions, means of the 100 repetitions were computed. A term not occurring in one repetition received a VIM of zero. Additionally, asymptotic 95% confidence intervals for these means μ were calculated by $\hat{\mu} \pm 1.96 \cdot \widehat{\text{se}}$, where $\widehat{\text{se}}$ is the estimated standard error. For binary outcomes, the AUC was used for determining VIMs, while for continuous outcomes, the MSE was employed.

Figure 7 depicts the determined VIMs. For all four scenarios and all three considered measures, the true influential input variables SNP1D, SNP2D, SNP3D receive the highest importance values. Theoretically non-influential terms comprised of variables not influencing the outcome were assigned importance values close to zero in all cases. In the first simulation scenario, the logic VIM and the removal VIM both assign the triplet SNP1D $\wedge$ SNP2D $\wedge$ SNP3D^c the highest importance among all interactions. The permutation VIM favors the sub-conjunction SNP2D $\wedge$ SNP3D^c of this triplet. Both interpretations are correct regarding the true model in their own sense, since the term SNP2D $\wedge$ SNP3D^c interacts with SNP1D due to squaring the linear predictor.

In the remaining three scenarios, all VIMs assign the term SNP2D $\wedge$ SNP3D^c the highest importance among all interactions. However, in the third scenario considering, as in the first scenario, the square of the linear predictor, the conjunction SNP1D $\wedge$ SNP2D $\wedge$ SNP3D^c and additionally sub-conjunctions receive importance values greater than zero. In the last scenario considering a continuous outcome and an influential environmental covariable, the interaction SNP2D $\wedge$ SNP3D^c received the highest importance overall for all three importance measures.

In the first three scenarios, the three single input variables yield the highest importances. This is due to the fact that the VIM of single input variables coincides with the standard definition of VIMs, i.e., the difference in error when informatively removing a single input variable. Thus, the VIM of a single input variables captures all of its effects, including effects of interaction in which this input variable participates. In the fourth scenario, the two-way interaction SNP2D $\wedge$ SNP3D^c seems to be identified in almost every logicDT application so that the single input variables SNP2D and SNP3D receive lower importances due to being identified less often. Hence, the importances should be compared in groups corresponding to the interaction order, i.e., marginal importances should be compared to each other, two-way interactions should be compared to each other, and so forth.

In summary, all measures yield very similar and plausible results. The determination of the logic VIM is considerably faster than the determination of the removal VIM and the permutation VIM, since the model does not have to be refitted and predictions do

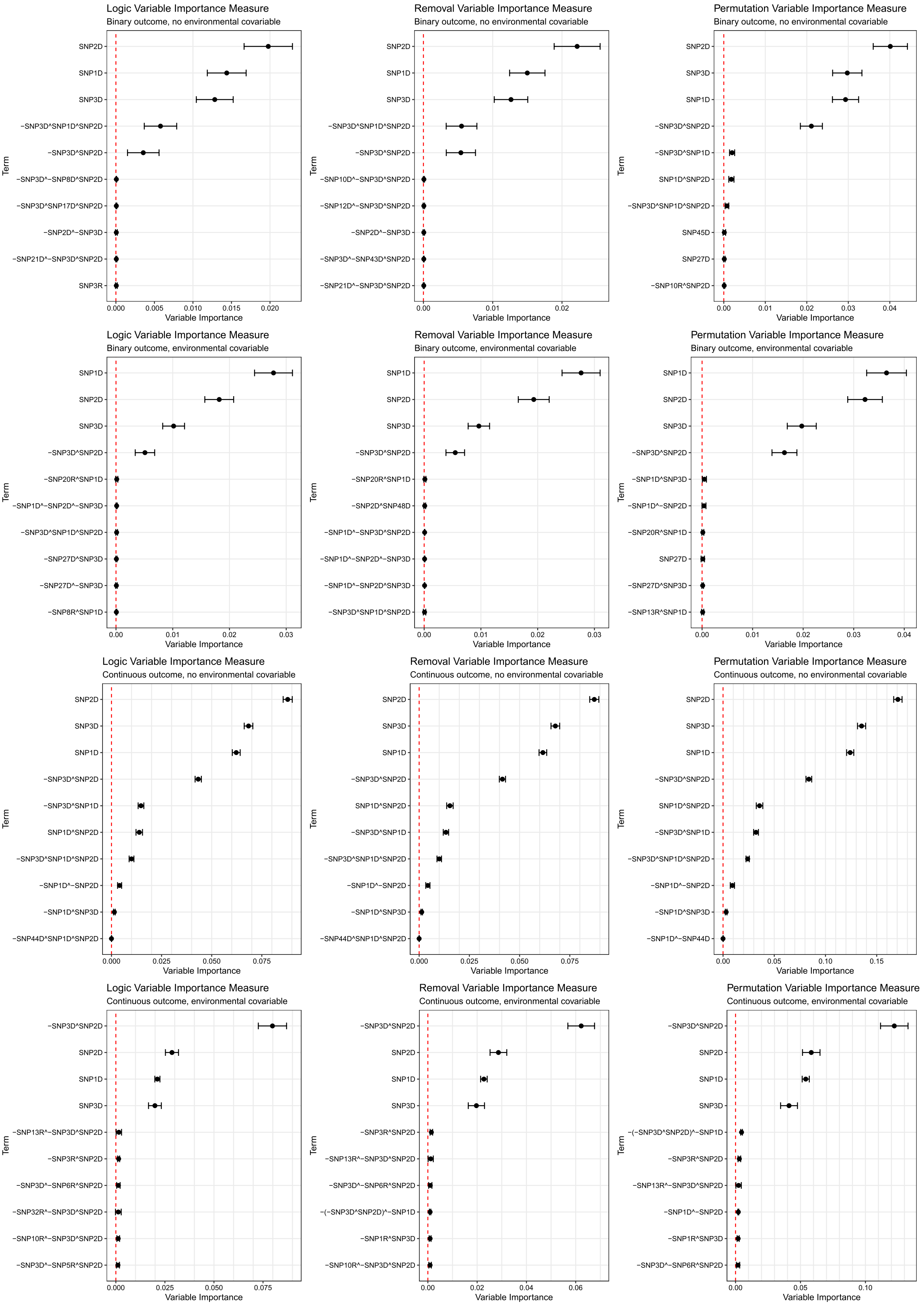

Fig. 7 Logic, removal, and permutation VIMs yielded by bagged logicDT models for the four scenarios in the simulation study. Adjustment for interactions and conjunctions was performed. Means and asymptotic 95% confidence intervals for the 100 repetitions are presented. Negations of input variables are denoted using a minus sign in the front

not have to be performed for a high number of permutations for computing the logic VIM. Instead, for a term consisting of k variables, only 2^k predictions have to performed and compared to the original prediction.

5.1.5 Second simulation setup

To investigate if logicDT is also suitable in scenarios in which a larger amount of input variables is considered and more input variables influence the outcome, we evaluate logicDT and the comparable methods in additional simulations. Two scenarios are investigated, one considering a binary outcome and one considering a continuous outcome, that are both simulated according to the linear model

$$\begin{aligned} g(\mathbb{E}[Y \mid \boldsymbol{X}, E]) = & -0.25 + \log(2) \cdot \mathbb{1}(\mathrm{SNP}_1 > 0) + \log(2.5) \cdot \frac{E}{20} \cdot \mathbb{1}(\mathrm{SNP}_2 > 0) \\ & - \log(1.5) \cdot \mathbb{1}(\mathrm{SNP}_3 = 2) - \log(1.5) \cdot \mathbb{1}(\mathrm{SNP}_4 = 0) \\ & + \log(3) \cdot \frac{E}{20} \cdot \mathbb{1}(\mathrm{SNP}_5 > 0) \cdot \mathbb{1}(\mathrm{SNP}_6 = 2) \\ & - \log(3) \cdot \mathbb{1}(\mathrm{SNP}_7 > 0) \cdot \mathbb{1}(\mathrm{SNP}_8 = 0) \cdot \mathbb{1}(\mathrm{SNP}_9 < 2), \end{aligned} \tag{12}$$

where g is the logit function for the binary outcome and the identity function for the continuous outcome. This model was chosen, since it exhibits a more complex structure, as nine SNPs influence the outcome as main effects, two-way interactions, three-way interactions, or gene-environment interactions. In total, 1000 SNPs (i.e., 2000 binary input variables coding for dominant and recessive modes of inheritance for these SNPs) and one continuous covariable were simulated for data sets with sample size $n = 1000$. The input variables are simulated analogously to the ones in Sect. 5.1.1. Both scenarios are, again, evaluated based on 100 independent replications, i.e., 100 random data sets, which are analogously to Sect. 5.1.1 divided into training, validation, and test data sets.

5.1.6 Predictive performance

In Fig. 8, the predictive performance of logicDT and the comparable methods in the application to the two additional simulation scenarios are depicted. Both scenarios seem to be relatively complex, since the discrepancy between the predictive performance of the true model and the fitted models is larger than, e.g., in the previously conducted simulations.

For the binary outcome, the best performance is induced by gradient boosting, closely followed by logicDT, bagged logicDT, random forests, logic regression, and logic bagging. Out of these methods, logicDT and logic regression are the only methods that yield interpretable models. Conventional decision trees, DL8.5, MOB, and RuleFit lead to lower AUCs.

For the continuous outcome, the best results are induced by logicDT, bagged logicDT, gradient boosting, logic regression, logic bagging, and RuleFit. The other interpretability-focused methods, namely conventional decision trees and MOB, yield lower predictive performances.

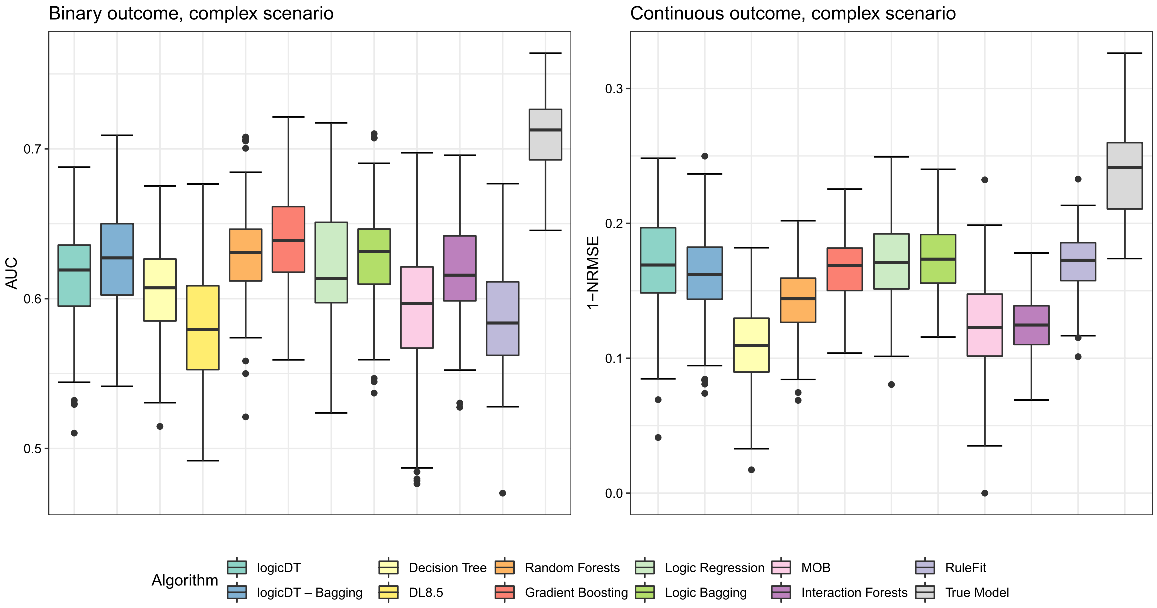

Fig. 8 Predictive performances of logicDT and the comparable methods in the simulation study considering two more complex scenarios. The performance for the binary outcome is measured by the AUC and the performance for the continuous outcome is measured by the complement of the NRMSE (normalized root mean squared error)

Hence, logicDT seems to be also applicable and yielding comparatively high predictive performances, when considering scenarios with larger numbers of input variables (here, 2000 binary input variables) and influential input variables.

5.1.7 Variable importance

In Fig. 9, the estimated variable importances by bagged logicDT in the application to the two additional simulation scenarios are displayed. Since a relatively complex scenario is considered, not every influential term is identified. Nonetheless, for the binary outcome and each considered VIM type, each term with a strongly positive variable importance is truly influential in the underlying data-generating model (12). Moreover, for both the binary and the continuous outcome and all VIM types, the two-way interaction $\text{SNP8D}^c \wedge \text{SNP7D}$ is correctly identified.

For the continuous outcome and the permutation VIM, the five top-ranking importances correspond to truly influential terms. However, the terms showing the next highest importances corresponding to theoretically non-influential terms such as $(\text{SNP8D}^c \wedge \text{SNP7D})^c \wedge \text{SNP2D}$ indicate that these terms are influential as well due to their importance confidence intervals fully being above zero. This issue of falsely identified terms seems to be alleviated when employing the logic VIM or the removal VIM due to less non-influential terms that yield VIM confidence intervals fully above zero when using these VIMs. This, thus, indicates that the logic VIM and the removal VIM in conjunction with the adjustment for interactions can also be employed in more complex scenarios with a larger number of input variables.

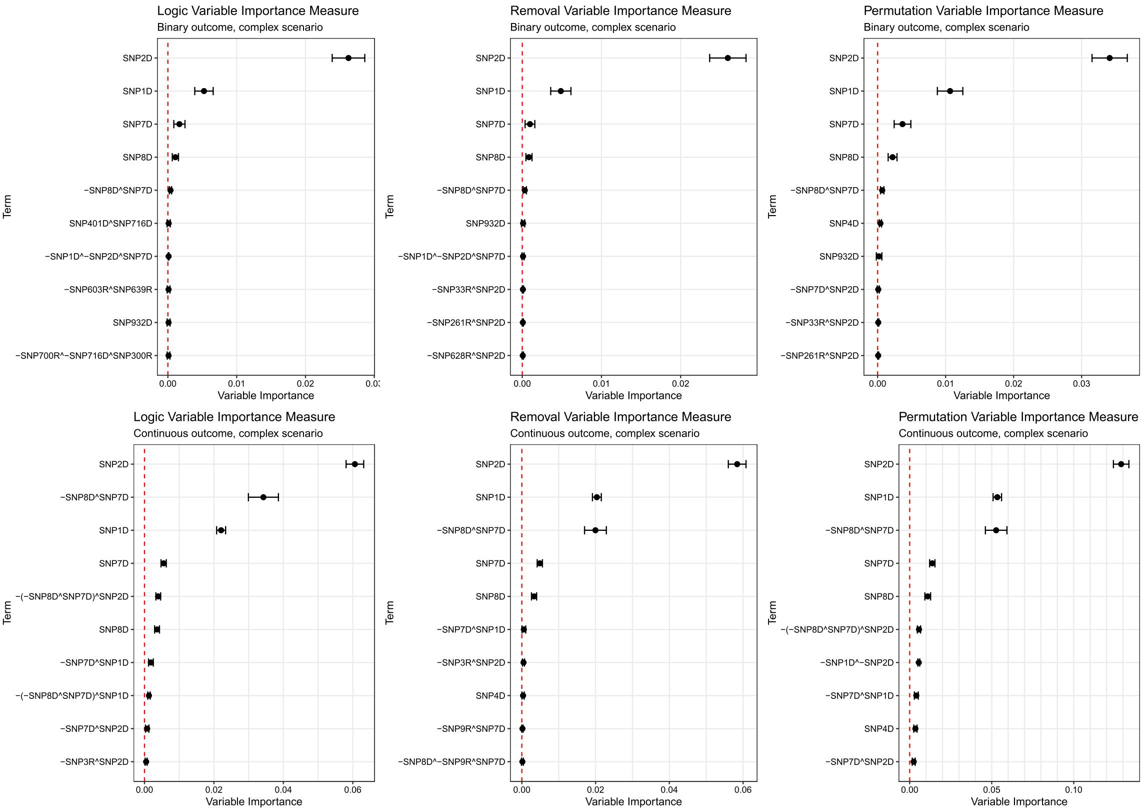

Fig. 9 Logic, removal, and permutation VIMs yielded by bagged logicDT models for the two more complex scenarios in the simulation study. Adjustment for interactions and conjunctions was performed. Means and asymptotic 95% confidence intervals for the 100 repetitions are presented. Negations of input variables are denoted using a minus sign in the front

5.2 Real data application

We have also applied logicDT and the comparable statistical learning methods to several real data sets, from which the data set of the SALIA study is of particular interest. Therefore, we consider, first, in the following subsections this study and the performance of logicDT and the comparable methods in their application to the data from the SALIA study. Afterwards, we summarize the results of the analyses of the other data sets in Sect. 5.2.4. A more detailed discussion of these evaluations can be found in Appendix 4.

5.2.1 SALIA study

logicDT was applied to a real data set from a German cohort study called the SALIA study (**S**tudy on the Influence of **A**ir Pollution on **L**ung, **I**nflammation and **A**ging, Schikowski et al., 2005). The results of logicDT were compared to the results of the methods also considered in the comparisons in Sect. 5.1. The data set consists of data from 517 women, from which 123 had a rheumatic disease so that 394 women did not show a rheumatic disease. For these women, data from 77 SNPs from the HLA-DRB1 gene, which presumably plays a major role in the heritability of rheumatoid arthritis (Clarke & Vyse, 2009), are available. For more details about the SALIA study itself and

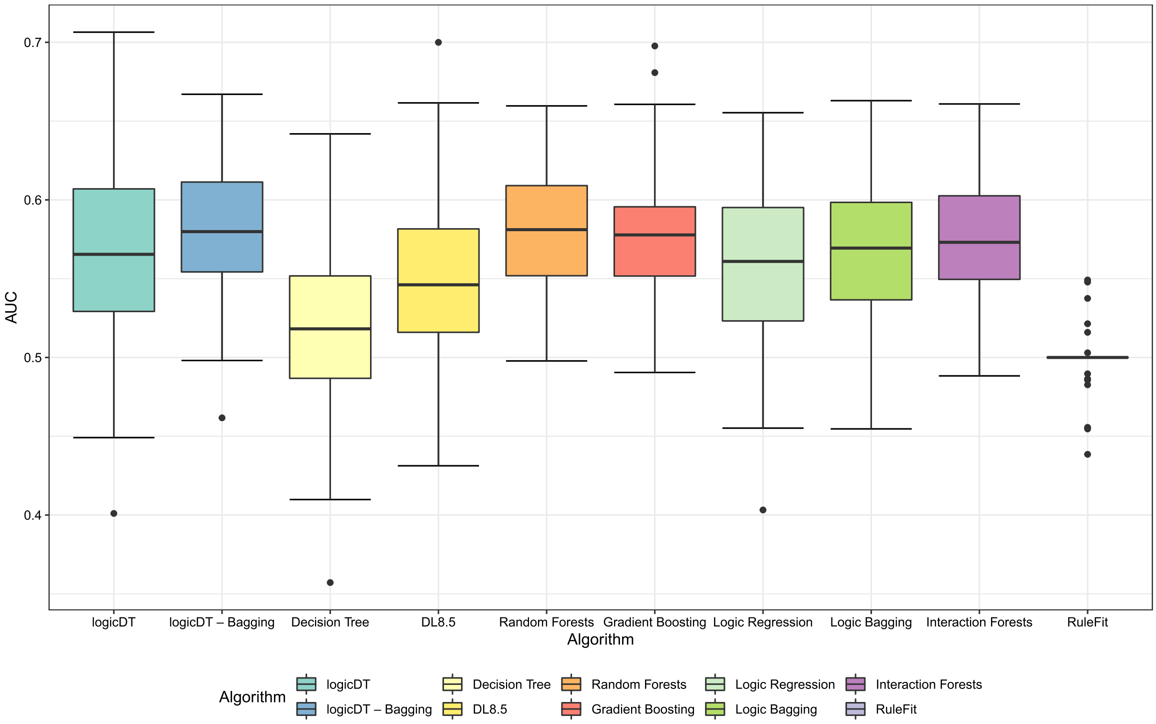

Fig. 10 Predictive performances of logicDT and the comparable methods in the evaluation of the SALIA data

an analysis of rheumatic diseases in the SALIA study, see Krämer et al. (2010) and Lau et al. (2022), respectively.

The analysis was performed using a similar scheme as in the simulation study. For 100 independent repetitions, training, validation and test data sets were randomly drawn from the total data set. Hyperparameter optimization was performed using, again, the parameter values summarized in Table 1.

5.2.2 Predictive performance

In Fig. 10, the performances of logicDT and the comparable methods in their application to the SNP data from the SALIA study are shown. This figure reveals that all evaluated statistical learning procedures induce similarly high AUCs, except for conventional decision trees, DL8.5, and RuleFit, which show inferior predictive performances. RuleFit seems to have issues to detect a signal in the data set at all, despite optimizing its hyperparameters.

We would like to point out that logicDT is the only other procedure than conventional decision trees, DL8.5, logic regression, and RuleFit that yields easily interpretable prediction models. In contrast to these models, logicDT still leads to comparatively high predictive performances. Single logic regression models yield similar AUCs as logicDT. However, due to logic regression models including complex terms consisting of mixtures of Boolean conjunctions and disjunctions, logic regression models tend to be harder to interpret than logicDT models.

Figure 11 shows the fitted logicDT model on the complete SALIA data. This tree is still relatively easy to interpret, i.e., it is easy to understand how predictions are made and which interactions are involved in the prediction. In comparison, the fitted logic regression model on the complete SALIA is given by

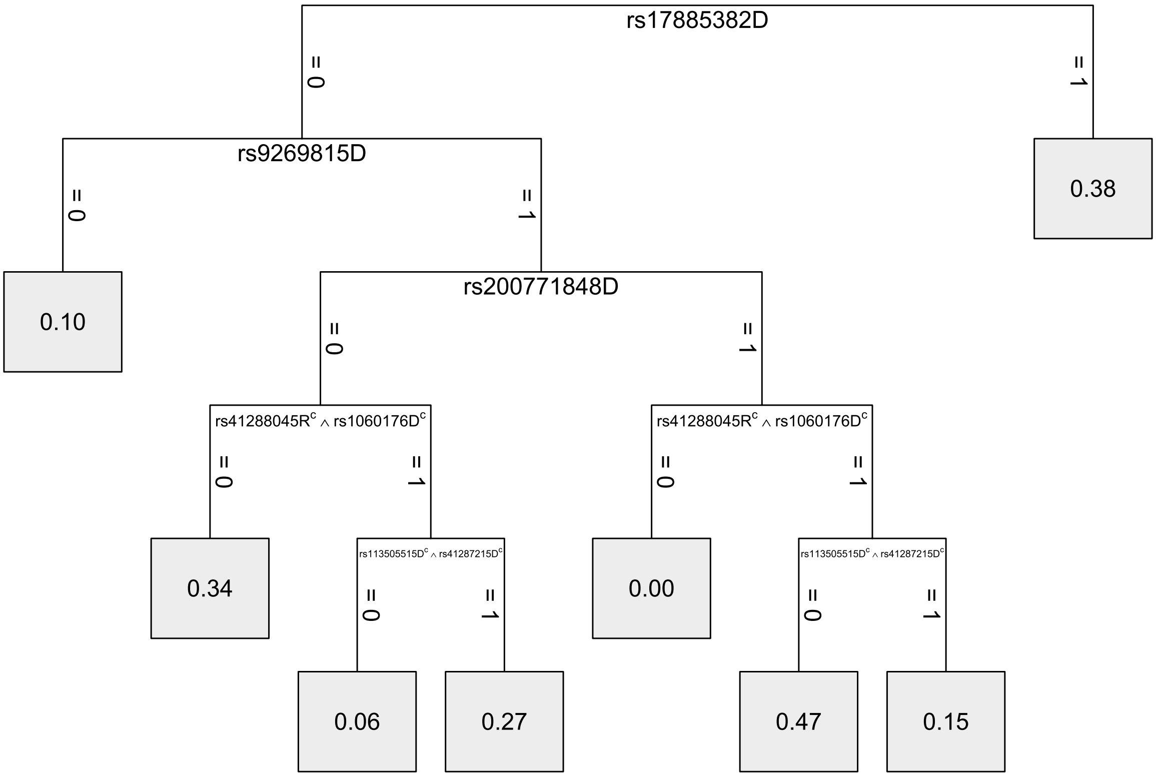
rs17885382D
= 0
= 1
0.38
rs9269815D
= 0
= 1
0.10
rs200771848D
= 0
= 1
rs41288045R^c ∧ rs1060176D^c
= 0
= 1
rs41288045R^c ∧ rs1060176D^c
= 0
= 1
0.34
rs113505515D^c ∧ rs41287215D^c
= 0
= 1
0.00
rs113505515D^c ∧ rs41287215D^c
= 0
= 1
0.06
0.27
0.47
0.15

Fig. 11 Fitted logicDT model on the complete SALIA data

$$
\begin{aligned}
\operatorname{logit}(\mathbb{P}(Y=1 \mid X)) = &- 1.14 \\
&- 19.63 \cdot \mathbb{1}((\mathrm{rs113608847D} \wedge (\mathrm{rs113505515D}^c \vee \mathrm{rs9270143R})) \\
&\qquad \wedge (\mathrm{rs1060176D} \vee (\mathrm{rs28724138R}^c \wedge \mathrm{rs17884945R}^c))) \\
&- 2.91 \cdot \mathbb{1}((\mathrm{rs34578704D}^c \wedge \mathrm{rs34084957D}) \\
&\qquad \vee ((\mathrm{rs41288045R} \vee \mathrm{rs9269814D}^c) \vee \mathrm{rs72844253R})) \\
&+ 1.41 \cdot \mathbb{1}((\mathrm{rs113322920D} \vee \mathrm{rs36101847R}) \wedge \mathrm{rs17879702D}^c).
\end{aligned}
$$

For this model, it is not trivial which interactions are involved in the prediction and how predictions for $\mathbb{P}(Y=1 \mid X)$ are constructed.

5.2.3 Variable importance

Figure 12 illustrates the measured variable importances in the application to the SALIA data for the three proposed VIM approaches using bagged logicDT models. In the top row, the importances for the top 5 single input variables are depicted. In the second and third row, the importances for the top 5 two-way and three-way interactions are shown, respectively.

For verifying whether the terms identified by logicDT really have an influence on the outcome of interest, i.e., the rheumatic disease status, we considered for each identified term X in Fig. 12 a logistic regression model

$$\operatorname{logit}(\mathbb{P}(Y=1 \mid X=x)) = \beta_0 + \beta_1 x \tag{13}$$

and performed statistical hypothesis tests testing whether the respective term has an influence on the outcome, i.e., testing $H_0 : \beta_1 = 0$ vs. $H_1 : \beta_1 \neq 0$ using a Wald test. For each

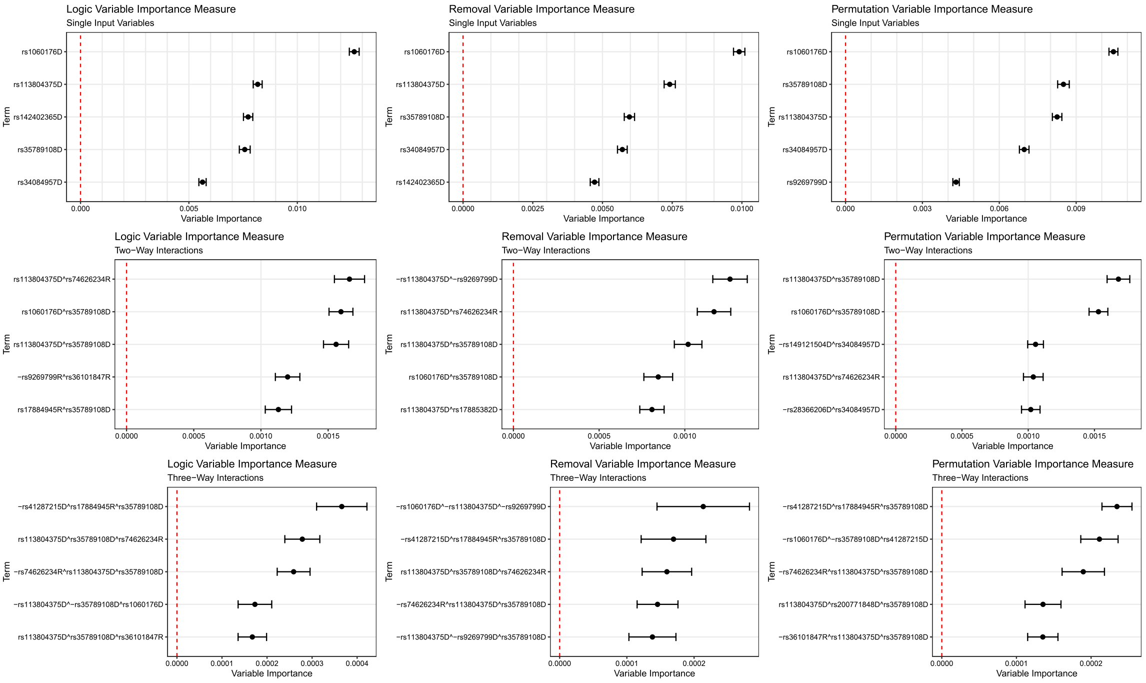

Fig. 12 Logic, removal, and permutation VIMs yielded by bagged logicDT models in the evaluation of the SALIA data—divided into VIMs of single input variables, two-way interactions and three-way interactions. Adjustment for interactions and conjunctions was performed. Means and asymptotic 95% confidence intervals for the 100 repetitions are presented. Negations of input variables are denoted using a minus sign in the front

Table 2 Numbers of identified terms from Fig. 12 that were significant with respect to $\alpha = 5\%$ using a false discovery rate adjustment

Significant terms/5	Logic VIM	Removal VIM	Permutation VIM
Single Input Variables	0	0	0
Two-Way Interactions	4	2	2
Three-Way Interactions	5	3	4

set of five identified terms, we evaluated how many terms lead to significant coefficients in the model from Eq. (13) using a significance level of $\alpha = 5\%$ and adjusting for multiple testing using the method by Benjamini and Hochberg (1995).

Table 2 shows the results for this post-hoc analysis. None of the identified single input variables proves to be significant. However, for the logic VIM, four of the five identified two-way interactions and all five three-way interactions seem to have a significant influence on the outcome. For the more computationally intensive removal and permutation VIMs, the results seem to be inferior, since only two of the five two-way interactions are significant, and three or four of the five three-way interactions, respectively, are significant.

Note that the VIMs of the single input variables depicted in Fig. 12 are considerably higher than the VIMs of the interaction terms, yet the single input variables were not significant. As discussed in the simulation study in Sect. 5.1.4, this is due to the fact that the VIMs for single input variables also capture the importance of interactions that contain the input variable of interest. Thus, if a single input variable is part of many interactions, this inflates its importance value without leading to a significant main effect of the variable. For example, the most influential input variable across all three VIM calculation approaches, rs1060176D, is in every considered situation part of one identified interaction term.

5.2.4 Additional real data evaluations

logicDT and the comparable methods are also evaluated in additional experiments using 24 real data sets from various application fields. The main result is that logicDT induces high predictive performances among single-model procedures in the application to these additional real data sets. Among the ensemble methods, bagged logicDT also induces for most data sets relatively high predictive performances. More details on the analyses of the additional real data sets can be found in Appendix 4.

6 Discussion

In this article, we have presented a statistical learning procedure called logicDT that is specifically tailored to finding interactions between binary input variables and that can also take continuous covariables into account by fitting regression models in the decision tree branches. In contrast to, e.g., logic regression, all possible interactions of the binary input data with this continuous covariable can be included in the prediction model as well as interactions between interactions of the binary input data. logicDT is aimed at maximizing both predictive power and interpretability motivated by applications in genetic epidemiology.

As a simulation study as well as real data applications show, logicDT is able to fulfill these objectives and yields comparable or better predictive performances as similar methods, while maintaining interpretability, which is lost when applying most other approaches. Moreover, through simulated annealing and theory on decision trees, theoretical success of logicDT, i.e., that the true underlying regression function is asymptotically attained, could be proven.

For maximizing the predictive performance regardless of being able to interpret how exactly predictions are made, bagging can be applied to logicDT, yielding performances as state-of-the-art algorithms such as random forests or gradient boosting.

Through different VIMs and VIM adjustment approaches for measuring the importances of interactions and specific conjunctions, highly predictive bagged logicDT models are still very useful for deriving which variables influence the outcome in interaction with which other variables. In comparison to standard VIM approaches, the proposed interaction VIM is able to capture influences of interactions and is not restricted to single input variables. Note that the proposed VIM adjustment approaches can also be applied to other statistical learning procedures, e.g., black-box methods such as deep neural networks or random forests, since no restricting assumptions on the model fitting procedure itself are made in these approaches.

Fitting logicDT models is computationally intensive due to the global search via simulated annealing, and takes, in particular, more time than fitting conventional decision trees that employ a greedy algorithm. However, as could be seen in the simulation study and the real data applications, logicDT consistently outperformed conventional decision trees considering the predictive performance. Moreover, logicDT still does not seem to be slower than other interpretability-focused methods such as logic regression or RuleFit. A model fitting time evaluation of logicDT and other procedures in the simulation study and real data application can be found in Appendix 5.

logicDT was designed for interpretable modeling in low- to mid-dimensional problems, e.g., considering single genes, pathways, or selections of SNPs that were significantly influencing the outcome in prior analyses. However, in theory, logicDT can be applied to problems with an arbitrarily large number p of input variables. Nonetheless, as shown in Sect. 3.8, the computational complexity of logicDT is polynomial in p under certain assumptions. Moreover, in practice, only finitely many computational resources are available. In simulations considering 1000 SNPs (i.e., $p = 2000$ input variables due to splitting each SNP into two binary variables) and a more complex underlying model, logicDT still induced relatively high predictive performances (see Sect. 5.1.5). Hence, we recommend applying logicDT in situations with $p \leq 2000$. For comparison, in the software implementation of logic regression, where also a stochastic search algorithm is employed, the authors allow a maximum of $p = 1000$ input variables (Kooperberg & Ruczinski, 2022).

The main issue of conventional decision trees is its instability issue, i.e., that small modifications of the training data set imply unproportionally severe alterations of the fitted model. This behavior is mainly induced by the greedy fitting algorithm (Li & Belford, 2002; Murthy & Salzberg, 1995). logicDT aims at identifying the globally optimal set of predictors and interactions responsible for the variation in the outcome. Thus, only important predictors are used for fitting the decision tree and interactions are already covered by single splits. Therefore, the instability issue should be diminished by logicDT.

The search procedure in logicDT utilizes the training data both for fitting decision trees and scoring them for guiding the search, which might suggest that this might lead to overfitting. However, both training trees based on states and evaluating states are part of the logicDT fitting procedure and the balance of overfitting and underfitting is controlled by the hyperparameters tuned using independent validation data (see Sect. 3.6). Moreover, established statistical modeling approaches such as stepwise linear regression or logic regression also employ the full training data set for both fitting the models and guiding the search. Nonetheless, one idea might be to further split the available training data into training data for fitting the decision tree based on the considered state and inner validation data for scoring the state's performance. However, due to the need for further splitting the available data, less observations are available for both the tree fitting step and the scoring step, leading to a decreased performance (on independent test data) compared to the original algorithm in empirical experiments (see Appendix 6). Moreover, the resulting model should not heavily rely on the data split used for this inner validation. Hence, ideally, multiple data splits—fitting and scoring multiple trees for one state and averaging the results as in (inner) cross-validation—should be used, leading to an increased computational burden.

Bagged logicDT was designed for situations in which a larger number of input variables influences the outcome or variable/interaction term importances shall be measured. In the simulation study conducted in Sect. 5.1.1, bagged logicDT performed similarly well compared to logicDT due to single logic decision trees being able to fully capture the considered underlying models. In additional simulations considering scenarios with larger numbers of influential input variables (see Sect. 5.1.5) and real data evaluations (see Appendix 4), bagged logicDT was able to achieve higher predictive performances in comparison to logicDT. Nevertheless, in these additional analyses, logicDT induced strong performances compared to other single-model methods.

For bagged logicDT, one idea to further increase its performance might be to further randomize the search similar to random forests. This could be realized by selecting a random sample of the neighbor states to be evaluated in each iteration of the greedy search, which is similar to randomly sampling potential splitting variables in random forests. However, this would create another hyperparameter—the number of randomly drawn candidate neighbor states—that potentially should be tuned and could depend on the total number of neighbor states that can change for each considered state.

logicDT is motivated by applications in genetic epidemiology in which mainly binary input data is analyzed. Although not considered in this article, it is possible to generalize logicDT to numerical input data by considering numerical interactions $\prod_j X_j$ instead of Boolean conjunctions $\bigwedge_j X_j$, where in the case of binary input data, these two definitions coincide.

The development of logicDT was, more precisely, motivated by the problem of constructing genetic risk scores that are usually built based on linkage-disequilibrium-based pruned SNPs, i.e., SNPs that can be interpreted as independent variables (So & Sham, 2017; Dudbridge & Newcombe, 2015). Therefore, throughout this manuscript, the assumption was made that there are no strong correlations between the considered input variables. In future research, logicDT and the interaction VIM could be analyzed and potentially adjusted for settings in which strong correlations between input variables exist so that, ideally, input variables (highly) correlated with truly predictive input variables do not diminish the importance of these truly predictive input variables.

If, additionally, a quantitative variable such as a quantitative environmental variable is considered, logicDT uses this covariable to fit regression models in the leaves of the decision tree. Since logicDT splits, in the context of genetic epidemiology, on genetic variants, a gene-environment is present if and only if the leaf regression models differ more than by fixed offsets describing marginal effects of the genetic variants. Thus, in future research, logicDT could be expanded for statistically testing the presence of a gene-environment interaction in the considered subregion of the DNA.

Moreover, the proposed interaction importance measuring methodology could also be expanded for statistically testing if certain single input variables or interaction terms significantly influence the outcome. This can, e.g., be used in the context of genetic epidemiology, testing the presence of gene-gene interactions. For implementing this testing procedure, the variable importance testing framework proposed by Watson and Wright (2021) might be applied to the importance measures proposed in this manuscript.

7 Conclusion

logicDT yields highly interpretable decision trees with superior predictive performances compared to other single-model procedures such as standard decision trees by being able to detect interaction effects between binary predictors on split level. Fitting ensembles of logicDT models through bagging can further increase the predictive performance if many predictors have effects on the outcome. The novel VIM adjustment procedure can be applied to these logicDT ensembles to derive which input variables influence the outcome in which interplay and magnitude—also measuring the importance of interaction effects between input variables.

Appendix 1: Simulated-annealing-based search procedure

The main methodology of logicDT, for which consistency is proven, employs simulated annealing as its search algorithm. In applications of logicDT, we suggest using an adaptive cooling schedule that requires no temperature tuning at all, which is in contrast to a geometric cooling schedule that is, e.g., used in logic regression. An adaptive cooling schedule automatically tunes the cooling behavior of simulated annealing, i.e., the start temperature, the temperature lowering steps or the Markov chain lengths, and the end temperature. Using an adaptive cooling schedule simplifies the application of simulated annealing, since these parameters do not have to be fine-tuned manually.

The start temperature is generally chosen such that at the beginning of the algorithm essentially a random walk is performed. For finding an appropriate initial temperature, a brief random walk over the state space (e.g., visiting 10,000 states) is carried out in logicDT and the state scores are recorded. Since the acceptance function

$$\gamma(\epsilon(s), \epsilon(s'), t) = \min\left\{1, \exp\left(\frac{\epsilon(s) - \epsilon(s')}{t}\right)\right\}$$

in simulated annealing is chosen so that better or equal states are automatically accepted, the temperature only influences the acceptance behavior of proposed worse states. Thus, only moves leading to worse states are used to estimate a temperature at which, e.g., 90% of the worse states are accepted.

We employ the homogeneous version of simulated annealing that runs through many consecutive homogeneous Markov chains. In practice, we limit the number of iterations per chain to, e.g., 1000 and adaptively choose the next temperature in a way that equilibrium of the next chain can be easily reattained. More precisely, we employ the temperature lowering scheme proposed by Huang et al. (1986) that is given by

$$t' = t \cdot \exp\left(-\lambda \frac{t}{\sigma(t)}\right),$$

where t is the current temperature, t' is the new temperature of the next Markov chain, and $\sigma(t)$ is the standard deviation of the scores observed in the finished Markov chain (see also, e.g., Van Laarhoven & Aarts, 1987). Here, $\lambda \in (0, 1]$ is a parameter controlling the speed of the total algorithm, which means that a higher value of λ leads to larger decreases in the temperature t, and hence, to less total iterations. Consequently, a value closer to 0 leads to a finer search, requiring more iterations. Generally, more iterations are preferable for approximating the theoretical asymptotic search. However, in practice, we recommend using a value of $\lambda \in [0.01, 0.1]$ for performing at least a few hundred thousand iterations.

For stopping the stochastic search, we evaluate the fraction of accepted states yielding a different score than the previous one, i.e., ignoring two neighbor states that yield the exact same score. If, e.g., for five consecutive chains the fraction of this adjusted state acceptance ratio is below 1%, the search is terminated. Alternatives include using the total number of chains instead of restricting to consecutive ones or using, similar to Triki et al. (2005), the standard deviation of the scores in a chain. For very small temperatures, simulated annealing should only move to better or equal states in terms of the score function. Thus, if an ideal state is reached, the score should no longer change, leading to a standard deviation of the score of 0.

Similar to the cooling schedule proposed by Triki et al. (2005), in the beginning of the search, the lowering of the temperature will also be triggered, if a threshold of accepted states in a single Markov chain is reached. This threshold might, e.g., be set to 50% of the total Markov chain length and prevents the search from focusing too long on the initial near random walk type of search, but instead focusing on the middle part of simulated annealing.

The theory of simulated annealing is based on two convergences, namely

- the convergence of the individual Markov chains, i.e., reaching equilibrium or the respective stationary distribution,
- the convergence of the temperature to 0, i.e., approaching an infinitesimal low temperature.

In practice, since limited computing resources are available, there is no guarantee that simulated annealing finishes in a global minimum. Thus, it might be possible that a globally optimal state is visited in the initial exploration of the state space, but due to relatively few iterations another local optimum is reached afterwards and is not abandoned anymore. We, therefore, let the algorithm also remember the best visited state so far in the search.

Due to noninformative terms or noninformative predictors in a conjunction, it might be possible that two neighbor states yield the exact same score. In this case, generally the simpler model is preferred. Thus, when the search is finished, each term is inspected for variables and conjunctions that do not improve the score and these variables or conjunctions are removed from the model. Furthermore, if a new neighbor is proposed that leads to exactly the same score as the current state, this new neighbor is accepted in simulated annealing, regardless of the current temperature.

Visiting a single state multiple times can also occur due to the random nature of simulated annealing itself. To account for this behavior in the searching procedure, a hash table containing sorted linked lists of the specific states and their respective scores is used for remembering already visited states. Thus, if a state is reached multiple times, the predictor transformation and the decision tree fitting do not have to be repeated.

Appendix 2: Consistency proof

In this appendix, we prove Theorem 1 that was stated in Sect. 3.7. For proving this theorem, some preliminary results are necessary that are proven in the following lemmata. We start by proving that simulated annealing leads to an optimal solution in logicDT.

Lemma 1 *The Markov chains constructed in logicDT fulfill the prerequisites of simulated annealing such that the stationary distributions* $\pi_t = \lim_{q \to \infty} \mathbb{P}(Q_t(q) = \cdot)$ *exist and it holds that*

$$\lim_{t \searrow 0} \pi_t(s) = \begin{cases} \dfrac{|\mathcal{R}_s|}{\sum_{s' \in \mathcal{R}_{\text{opt}}} |\mathcal{R}_{s'}|}, & s \in \mathcal{R}_{\text{opt}} \\ 0, & s \notin \mathcal{R}_{\text{opt}} \end{cases}$$

for the set $\mathcal{R}_s$ *of neighbor states of state s and the set* $\mathcal{R}_{\text{opt}}$ *of optimal states.*

Proof For establishing convergence of the individual (finite and homogeneous) Markov chains to their stationary distributions, it is sufficient to prove their irreducibility and aperiodicity (e.g., Theorem 1 in Section 3.1.2, Van Laarhoven & Aarts, 1987).

The Markov chains Q_t in simulated annealing are generally based on the transition probabilities

$$\tau\big(s, s', t\big) \ := \ \mathbb{P}\big(Q_t(q+1) = s' \mid Q_t(q) = s\big) \ = \ \gamma\big(\epsilon(s), \epsilon(s'), t\big) \cdot \beta\big(s, s'\big)$$

for $s \neq s'$ and all $q \in \mathbb{N}$, where $\gamma(\epsilon(s), \epsilon(s'), t)$ describes the acceptance probability depending on the scores $\epsilon(\cdot)$ of the states and $\beta(s, s')$ yields the generation probability of s' given state s. In logicDT, the standard acceptance function

$$\gamma(\epsilon(s), \epsilon(s'), t) \ = \ \min\left\{1, \exp\left(\frac{\epsilon(s) - \epsilon(s')}{t}\right)\right\} \tag{14}$$

is used together with the uniform distribution for the generation probability

$$\beta(s, s') \ = \ \begin{cases} \frac{1}{|\mathcal{R}_s|}, & s' \in \mathcal{R}_s \\ 0, & s' \notin \mathcal{R}_s. \end{cases} \tag{15}$$

Since $\gamma(\epsilon(s), \epsilon(s'), t) > 0$ for every pair of states s, s' and $t > 0$ and the choice of the moves, i.e., modifications of states, proposed in Sect. 3.2 ensure that each state can be reached from any other state in a finite number of steps, the Markov chains in logicDT are irreducible.

Aperiodicity is fulfilled, if for all states s the greatest common divisor (gcd) of

$$\left\{n \in \mathbb{N} \mid \tau_n(s, s, t) := \mathbb{P}(Q_t(1+n) = s \mid Q_t(1) = s) > 0\right\}$$

is equal to 1. This property would be directly fulfilled, if the chains would be reflexive, i.e., fulfilling $\tau(s, s, t) > 0$ for each state s. However, since it might be the case that a state has only neighbors exhibiting better scores, leading to $\gamma(\epsilon(s), \epsilon(s'), t) = 1$ for each $s' \in \mathcal{R}_s$, the probability of staying in state s can be equal to 0, as, for the probability of proposing the current state, it holds that $\beta(s, s) = 0$ by choice of β. Therefore, three different cases for states s have to be considered.

Case 1: s has a neighbor state s' with $\epsilon(s') < \epsilon(s)$. In this case, the probability $\tau(s, s', t)$ of changing to state s' is positive. The probability $\tau(s', s, t)$ of returning to s is positive as well, which is due to $\gamma > 0$. Furthermore, the probability $\tau(s', s', t)$ of remaining in s' is also positive, since, if s is generated by $\beta(s', \cdot)$, s will be accepted with probability $\gamma(s', s, t) < 1$ because of $\epsilon(s) - \epsilon(s') > 0$ and the choice of γ in Eq. (14). Thus, $\tau_2(s, s, t) > 0$ and $\tau_3(s, s, t) > 0$ hold true yielding the greatest common divisor of $\gcd(2, 3) = 1$.
Case 2: s has at least one neighbor state s' with $\epsilon(s') > \epsilon(s)$, but no neighbor s'' with $\epsilon(s'') < \epsilon(s)$. In this case, it holds that $\gamma(\epsilon(s), \epsilon(s'), t) \in (0, 1)$, and thus,

$$\tau_1(s, s, t) \ = \ \tau(s, s, t) \ > \ 0.$$

Case 3: For all neighbor states s' of s, it holds that $\epsilon(s') = \epsilon(s)$. Here, we have

$$\gamma(\epsilon(s), \epsilon(s'), t) \ = \ \gamma(\epsilon(s'), \epsilon(s), t) \ = \ 1,$$

and therefore, $\tau_2(s, s, t) > 0$. Let s'' be another state with $\epsilon(s'') \neq \epsilon(s)$. Such a state has to exist, since otherwise each state would have the exact same score. The state s'' can be chosen such that, due to the irreducibility, there exists a sequence of states $(s, s_1, s_2, \dots, s_n, s'')$, in which each succeeding state is a neighbor of its predecessor, with

$$\epsilon(s) = \epsilon(s_1) = \epsilon(s_2) = \cdots = \epsilon(s_n)$$

for any $n \in \mathbb{N}$. Thus, it follows $\epsilon(s_n) \neq \epsilon(s'')$.
Case 3.1: $\epsilon(s_n) > \epsilon(s'')$. Due to $\epsilon(s'') - \epsilon(s_n) < 0$ and Eq. (14), it follows $\gamma(s'', s_n, t) < 1$, and hence, $\tau(s'', s'', t) > 0$. Using the state sequence $(s, s_1, \dots, s_n, s'', s'', s_n, \dots, s_1, s)$, it becomes obvious that $\tau_{2n+3}(s, s, t)$ is positive. Furthermore, it follows that $\gcd(2, 2n + 3) = 1$, as $2n + 3$ is odd.
Case 3.2: $\epsilon(s_n) < \epsilon(s'')$. Analogously to Case 3.1, it follows that $\tau(s_n, s_n, t) > 0$. Using the state sequence $(s, s_1, \dots, s_n, s_n, \dots, s_1, s)$, the probability $\tau_{2n+1}(s, s, t)$ has to be positive so that $\gcd(2, 2n + 1) = 1$, since $2n + 1$ is odd.

Thus, aperiodicity is given so that the individual limiting distributions exist.

Applying Theorem 2 from Section 3.1.3 of Van Laarhoven and Aarts (1987) to the constructed Markov chains using the choices for γ in Eq. (14) and β in Eq. (15) directly shows that the stationary distributions converge to a distribution that exactly has the set of optimal states as its support. □

Now we have to show that the empirical risk minimization (ERM), which is performed by simulated annealing, is asymptotically equivalent to a true risk minimization in logicDT.

Lemma 2 (ERM consistency of logicDT) *Let the outcome Y be bounded. Then, logicDT is strongly consistent with respect to empirical risk minimization, i.e.,*

$$\sup_T \left| R_{\text{emp}}(T) - R_{\text{true}}(T) \right| \xrightarrow[n\to\infty]{\text{a.s.}} 0,$$

where $R_{\text{emp}}(T) = \frac{1}{n}\sum_{i=1}^n L(y_i, T(\boldsymbol{x}_i))$ *is the empirical risk,* $R_{\text{true}}(T) = \mathbb{E}_{(X,Y)}[L(Y, T(\boldsymbol{X}))]$ *is the true risk, and* $L(y, \hat{y}) = (y - \hat{y})^2$ *is the squared error loss.*

Proof By assumption, Y is bounded. Thus, as the predictions of decision trees are generated by means of observed values, the predictions are bounded by the same bound. Furthermore, the L_2 loss is bounded likewise. Let this bound be given by $B > 0$, i.e., $L(y, \hat{y}) \in [0, B]$.

In order to prove distribution-independent ERM consistency, it is necessary and sufficient that the VC (Vapnik and Chervonenkis) dimension is finite (Vapnik, 2000), where the VC dimension is defined as the maximum number m of data points $z_1, \dots, z_m := (\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_m, y_m)$ that can be shattered by a binary loss function $L(y, T(\boldsymbol{x})) \in \{0, 1\}$. For $m \in \mathbb{N}$, there thus exists a sample $z_1, \dots, z_m$ such that for all possible 2^m binary outcomes $\in \{0, 1\}^m$ there exists a prediction function T in the considered space that divides the sample according to the label setting using the loss function $L(y, T(\boldsymbol{x}))$. In the general regression setting, the VC dimension is defined as the VC dimension of the indicators $\mathbb{1}(L(y, T(\boldsymbol{x})) \geq \beta)$, where $\beta \in [0, B]$ is interpreted as part of the function space for the determination of the VC dimension so that for each outcome setting a function T and a value for β have to be found.

For deriving the VC dimension of logicDT, note that the prediction values for each predictor setting can be chosen independently, i.e., it is only necessary to consider for how many data points the data can be shattered along one single predictor setting. In the case of not fully grown trees with shared leaves for different possible predictor settings (for example, a tree stump only splitting on $X_1 \in \{0, 1\}$ such that $T((X_1, 0)) = T((X_1, 1))$), the

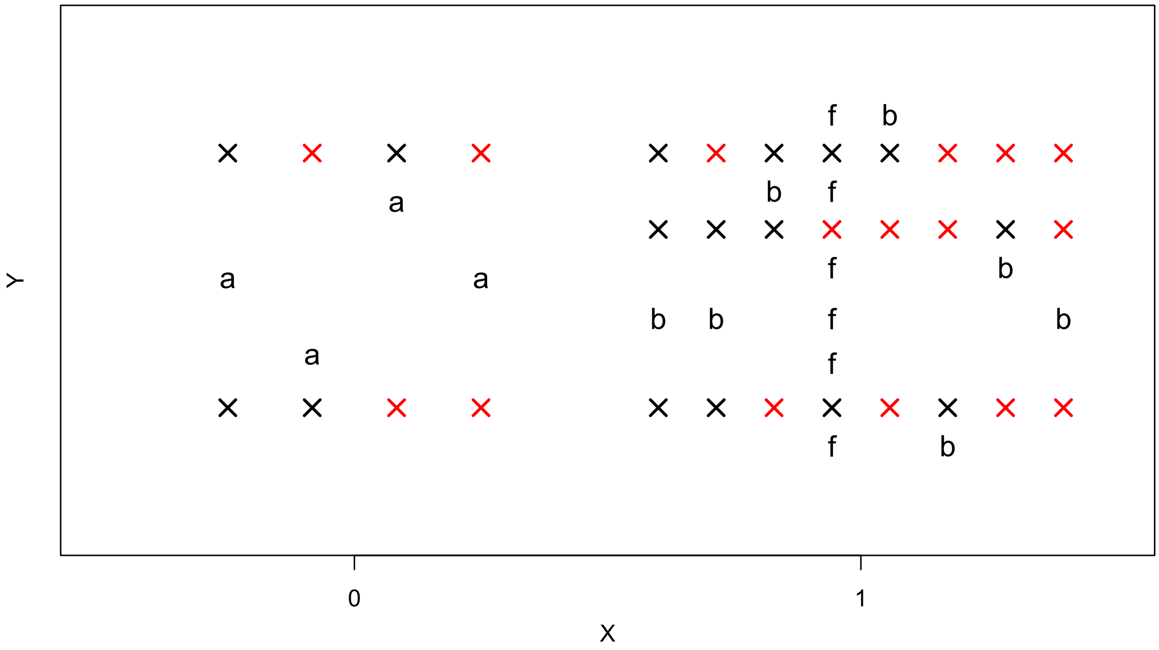

Fig. 13 VC dimension illustration for logicDT models. Here, one binary predictor $X \in \{0, 1\}$ is considered. For $X = 0$, all $2^2 = 4$ classifications for two data points are depicted. For $X = 1$, all $2^3 = 8$ classifications for three data points are shown. Black crosses indicate $\mathbb{1}(L(y, T(\boldsymbol{x})) \geq \beta) = 0$. Red crosses indicate $\mathbb{1}(L(y, T(\boldsymbol{x})) \geq \beta) = 1$. a and b are the (fixed) predictions $T(0) = a$, $T(1) = b$ such that the corresponding classification pattern can be achieved, i.e., there exists an appropriate β. f depicts the situation in which an appropriate prediction value, and thus, an appropriate tree cannot be constructed

prediction values are not necessarily independent of each other. However, in this case, the number of shatterable data points decreases compared to the independent prediction case so that this case does not have to be considered with regard to the VC dimension. Thus, it is sufficient to consider one single predictor $X \in \{0, 1\}$, since the shattering behavior only has to be analyzed independently for each setting.

Figure 13 depicts the shatterability for two and three data points, respectively. Two observations on one axis, as depicted here for $X = 0$, can be shattered by properly positioning the corresponding prediction value $a = T(0)$ and choosing an adequate β so that

$$\mathbb{1}((y-a)^2 \geq \beta) \;=\; \begin{cases} \text{Red} \times, & (y-a)^2 \geq \beta \\ \text{Black} \times, & (y-a)^2 < \beta, \end{cases}$$

i.e., choosing a and β such that red crosses are "far away" from a and black crosses are "close" to a.

For three data points, there is only one problematic labeling: If three different observations are considered that lie on one axis, one data point has to be the middle point. This middle point cannot be classified as 1/red while classifying the outer points as 0/black. This is due to the fact that the middle point needs to be far away from the prediction $b = T(1)$ to achieve this labeling, while the surrounding points need to be close to b, which is not possible.

Thus, for each prediction axis/tree branch, two is the maximum number of points that can be shattered. Since for p predictors there are 2^p possible predictor settings and two data points can be shattered for each setting, the VC dimension $\mathcal{VC}$ of logicDT is equal to

$$\mathcal{VC} \;=\; 2 \cdot 2^p \;=\; 2^{p+1}.$$

For bounded loss composition functions $L\circ\tilde{T} : \mathcal{X}\times\mathcal{Y}\to[0,B]$ with $\tilde{T}(\boldsymbol{x},y) := (y,T(\boldsymbol{x}))$, where L is in here given by $L(y,\hat{y}) = (y-\hat{y})^2$ so that $(L\circ\tilde{T})(\boldsymbol{x},y) = (y-T(\boldsymbol{x}))^2$, a uniform bound

$$\mathbb{P}\left(\sup_T \left|R_{\text{emp}}(T) - R_{\text{true}}(T)\right| > \varepsilon\right) \le 4\exp\left\{\left(\frac{G(2n)}{n} - \frac{\varepsilon^2}{B^2}\right)n\right\} \tag{16}$$

involving the growth function G holds for all $\varepsilon > 0$ (see Equation (3.10), Vapnik, 2000). This growth function G is bounded by a function of the VC dimension. In particular, for $n > \mathcal{VC}$, it holds that

$$G(n) \le \mathcal{VC}\left(\log\left(\frac{n}{\mathcal{VC}}\right) + 1\right) \tag{17}$$

(see Equation (3.23), Vapnik, 2000).

For proving almost sure convergence of (16), i.e., for proving

$$\sup_T \left|R_{\text{emp}}(T) - R_{\text{true}}(T)\right| \xrightarrow[n\to\infty]{\text{a.s.}} 0, \tag{18}$$

it suffices to show that the corresponding series converges (see, e.g., Corollary 1, Section 1.11.1, Vapnik, 1998), i.e., that

$$\sum_{n=1}^{\infty} \mathbb{P}\left(\sup_T \left|R_{\text{emp}}(T) - R_{\text{true}}(T)\right| > \varepsilon\right) < \infty.$$

Using (17), the right-hand side of (16) is bounded by

$$4\exp\left\{\left(\frac{G(2n)}{n} - \frac{\varepsilon^2}{B^2}\right)n\right\} \le 4\exp\left\{\mathcal{VC}\left(\log\left(\frac{2n}{\mathcal{VC}}\right) + 1\right) - \frac{\varepsilon^2}{B^2}n\right\}.$$

Using the ratio test for checking the convergence of series, the ratio of two consecutive summands is given by

$$\begin{aligned}
\mathcal{R}_n^{n+1} :&= \frac{4\exp\left\{\mathcal{VC}\left(\log\left(\frac{2(n+1)}{\mathcal{VC}}\right)+1\right) - \frac{\varepsilon^2}{B^2}(n+1)\right\}}{4\exp\left\{\mathcal{VC}\left(\log\left(\frac{2n}{\mathcal{VC}}\right)+1\right) - \frac{\varepsilon^2}{B^2}n\right\}} \\
&= \frac{\exp\left\{\mathcal{VC}\left(\log\left(\frac{2(n+1)}{\mathcal{VC}}\right)\right)\right\}}{\exp\left\{\mathcal{VC}\left(\log\left(\frac{2n}{\mathcal{VC}}\right)\right)\right\}} \exp\left\{-\frac{\varepsilon^2}{B^2}\right\} \\
&= \frac{\left(\frac{2(n+1)}{\mathcal{VC}}\right)^{\mathcal{VC}}}{\left(\frac{2n}{\mathcal{VC}}\right)^{\mathcal{VC}}} \exp\left\{-\frac{\varepsilon^2}{B^2}\right\} = \underbrace{\left(\frac{n+1}{n}\right)^{\mathcal{VC}}}_{\searrow\, 1 \text{ as } n\to\infty} \underbrace{\exp\left\{-\frac{\varepsilon^2}{B^2}\right\}}_{<\,1 \text{ fixed}}.
\end{aligned}$$

Thus, for this ratio $\mathcal{R}_n^{n+1}$, it follows that there exists a number $\tilde{n}\in\mathbb{N}$ so that for all $n > \tilde{n}$ it holds $\mathcal{R}_n^{n+1} < 1$. Therefore, the series converges and almost sure convergence in (18) is established. □

Now it has to be shown that the true regression function can be fitted by logicDT so that the risk minimizing logicDT function asymptotically becomes the true regression function.

Lemma 3 (Each model is possible in logicDT) *Let $\mu : \{0,1\}^p \to \mathcal{Y}$ be a p-dimensional regression function with $\mathcal{Y} \subseteq \mathbb{R}$. Then, μ can be fitted by logicDT, i.e., $\mu \in \mathcal{L}$ with $\mathcal{L}$ being the class of all logicDT models.*

Proof Since μ takes only binary predictors as its input, μ can be expressed as

$$\mu(\boldsymbol{X}) \;=\; g_0 + \sum_{j=1}^{m} g_j \cdot \mathbb{1}\left(X_{k_{j,1}}^{(c)} \wedge \cdots \wedge X_{k_{j,p_j}}^{(c)} \right)$$

for values $g_0, g_j \in \mathcal{Y}$ and distinct conjunctions $C_j(\boldsymbol{X}) := X_{k_{j,1}}^{(c)} \wedge \cdots \wedge X_{k_{j,p_j}}^{(c)}$, where these conjunctions are distinct in the sense that for a given $\boldsymbol{x} \in \{0,1\}^p$ it holds that $\mathbb{1}(C_j(\boldsymbol{x})) = 1$ is true for at most one $j \in \{1, \dots, m\}$. Let $\mathcal{D}_n$ be a noise-free training data set that fully resembles μ, i.e.,

$$\begin{aligned}\mathcal{D}_n \;\subseteq\; \Big\{ (\boldsymbol{x}, y) :\quad & \Big[y = g_0 + g_j \wedge j \neq 0 \wedge \mathbb{1}(C_i(\boldsymbol{x})) = \mathbb{1}(i = j)\ \forall i \Big] \\ & \vee \Big[y = g_0 \wedge \mathbb{1}(C_i(\boldsymbol{x})) = 0\ \forall i \Big] \Big\}\end{aligned}$$

with the additional restriction that each conjunction scenario C_j and the null scenario with $C_j = 0$ for all j have to occur at least once in $\mathcal{D}_n$. Using a proper logicDT state, i.e., a set of conjunctions that distinguish between the conjunctions that compose μ, the corresponding fitted logic decision tree assigns the ideal values g_0 or $g_0 + g_j$ to its leaves. Thus, the resulting model is equal to μ. □

Now the lemmata can be assembled for proving Theorem 1.

Proof of Theorem 1 Simulated annealing operates on a finite state space, which is also the case for logicDT. In logicDT, simulated annealing leads with probability 1 to an ideal model on the training data (see Lemma 1), i.e.,

$$\lim_{t \searrow 0} \lim_{q \to \infty} \mathbb{P}(Q_t(q) \in \mathcal{R}_{\text{opt}}) \;=\; 1$$

for a temperature $t \geq 0$, the homogeneous Markov chains Q_t, and the set of optimal states $\mathcal{R}_{\text{opt}}$. More specifically, the stationary distribution $\pi_t = \lim_{q\to\infty} \mathbb{P}(Q_t(q) = \cdot)$ converges for $t \searrow 0$ to a specific distribution on $\mathcal{R}_{\text{opt}}$, namely

$$\lim_{t \searrow 0} \pi_t(s) \;=\; \begin{cases} \frac{|\mathcal{R}_s|}{\sum_{s' \in \mathcal{R}_{\text{opt}}} |\mathcal{R}_{s'}|}, & s \in \mathcal{R}_{\text{opt}} \\ 0, & s \notin \mathcal{R}_{\text{opt}}, \end{cases} \tag{19}$$

where $\mathcal{R}_s$ is the set of neighbor states of state s. Thus, if this final stationary distribution is reached, an optimal model has to be attained due to the finiteness of the state space.

For proving consistency of random forests with respect to the number of observations, Scornet et al. (2015) studied a theoretical random forest with an infinite number of trees due to the pointwise almost sure convergence resulting from the law of large numbers. Similarly, we assume that the convergences in simulated annealing have occurred, and therefore, an empirical-risk-minimizing logicDT model is given due to the stationary distribution in Eq. (19). The CART methodology ensures that, for a given predictor/conjunction setting, the empirical-risk-minimizing decision tree is grown, if the tree is allowed to fully develop, i.e., not using any stopping criteria, since the prediction values are obtained by empirical risk minimization in the respective leaves (Breiman et al., 1984).

Note that this model will be in the set $\mathcal{R}_{\text{opt}}$ of empirical risk minimizing models, if the original predictor model consisting of the input variables $X_1, \dots, X_p$ is included in the considered state space. However, if the true underlying model μ is not a linear function of the individual predictors, the original model $\{\{X_1\}, \dots, \{X_p\}\}$ and equivalent extensions may be excluded from the state space while maintaining consistency. Thus, this theorem shows that logicDT models different from the original CART are consistent as long as the true function exhibits an adequate structure.

Let T_n be the empirical risk minimizer and μ be the true regression function. Applying Lemma 10.1 from Györfi et al. (2002) yields

$$\begin{aligned}\mathbb{E}_{(X,Y)}\left[(\mu(\boldsymbol{x}) - T_n(\boldsymbol{x}))^2\right] \;\leq\; & 2\sup_T \left| \frac{1}{n}\sum_{i=1}^{n}(y_i - T(\boldsymbol{x}_i))^2 - \mathbb{E}_{(X,Y)}\left[(Y - T(\boldsymbol{X}))^2\right] \right| \\ & + \inf_T \mathbb{E}_{(X,Y)}\left[(\mu(\boldsymbol{X}) - T(\boldsymbol{X}))^2\right],\end{aligned} \tag{20}$$

where the supremum and the infimum are determined over all logicDT models T.

Using Lemma 2, ERM consistency is established, i.e.,

$$\begin{aligned}\sup_T \left| R_{\text{emp}}(T) - R_{\text{true}}(T) \right| \;=\; & \sup_T \left| \frac{1}{n}\sum_{i=1}^{n}(y_i - T(\boldsymbol{x}_i))^2 - \mathbb{E}_{(X,Y)}\left[(Y - T(\boldsymbol{X}))^2\right] \right| \\ & \xrightarrow[n\to\infty]{\text{a.s.}} \; 0,\end{aligned}$$

where the almost sure convergence occurs with respect to the training data distribution $\mathbb{P}_{\mathcal{D}_n} = \mathbb{P}_{(X,Y)}^{\otimes n}$. Therefore, the first term on the right-hand side of (20) converges almost surely to zero.

Lemma 3 states that logicDT can lead to every possible regression function μ. Thus, it follows

$$\inf_T \mathbb{E}_{(X,Y)}\left[(\mu(\boldsymbol{X}) - T(\boldsymbol{X}))^2\right] \;=\; \mathbb{E}_{(X,Y)}\left[(\mu(\boldsymbol{X}) - \mu(\boldsymbol{X}))^2\right] \;=\; 0$$

so that the second term on the right-hand side of (20) vanishes.

Hence, in total, we obtain

$$\mathbb{E}_{(X,Y)}\left[(\mu(\boldsymbol{X}) - T_n(\boldsymbol{X}))^2\right] \;\xrightarrow[n\to\infty]{\text{a.s.}}\; 0,$$

which was to be shown. □

Appendix 3: Computational complexity proof

In this appendix, we prove Theorem 2 and Corollary 1 that were stated in Sect. 3.8.

Proof of Theorem 2 Following Algorithm 2, logicDT modifies the current state, creates a tree training data set, and fits and evaluates a decision tree based on this tree training data set to decide if the newly proposed state is accepted in every search iteration. Hence, the computational complexities of these individual steps have to be determined.

State modifications are performed randomly by modifying one variable in the current state. Therefore, the complexity of state modifications is given by $\mathcal{O}(1)$.

Tree training data sets are obtained by computing Boolean conjunctions using the variables in the considered term for each term in the considered state and each training observation (see Sect. 3.2). Since a state contains at most max_vars variables, transforming a training data set into a tree training data set amounts to a complexity of $\mathcal{O}(n \cdot \texttt{max\_vars})$.

Decision trees are fitted by recursively screening all p input variables for the best split (see Algorithm 1). This screening amounts to a complexity of $\mathcal{O}(np)$ for n training observations and p input variables and it is performed for at most $\left\lfloor \frac{n}{\texttt{nodesize}} \right\rfloor - 1$ inner nodes (corresponding to the worst-case scenario of an unbalanced tree in which the observations are perfectly divided into leaves of sample size nodesize). Thus, the (worst-case) complexity of fitting decision trees is given by $\mathcal{O}(n^2 p/\texttt{nodesize})$. This complexity remains valid for the case in which one additional continuous covariable is included due to univariate linear regression/LDA models being fitted and evaluated using closed-form solutions (i.e., each fitting/evaluation of these univariate regression models amounts to a complexity of $\mathcal{O}(n)$).

Since a maximum of max_conj input variables are used for fitting a logic decision tree, the tree fitting (and scoring) complexity in logicDT is given by $\mathcal{O}(n^2\texttt{max\_conj}/\texttt{nodesize})$. Therefore, using the aforementioned complexities, the computational complexity of logicDT is given by

$$\mathcal{O}\left(M\left[n \cdot \texttt{max\_vars} + n^2 \frac{\texttt{max\_conj}}{\texttt{nodesize}}\right]\right) = \mathcal{O}\left(Mn\left[\texttt{max\_vars} + \texttt{max\_conj}\frac{n}{\texttt{nodesize}}\right]\right),$$

which was to be shown. □

Proof of Corollary 1 The number M of search steps that are conducted in similar simulated-annealing-based search procedures is in the magnitude of $\mathcal{O}(L\log(|S|))$ (Van Laarhoven & Aarts, 1987), where L is the number of iterations performed per Markov chain and S is the search space. Since the search space considered in logicDT consists of sets of possible Boolean conjunctions that include at most max_conj conjunctions and at most max_vars input variables, the magnitude of this search space is given by

$$|S| \in \mathcal{O}((2p)^{\texttt{max\_vars}} \cdot \texttt{max\_conj}^{\texttt{max\_vars}}).$$

The first factor amounts for all selections of input variables or their negations of size max_vars, while the second factor amounts for the number of possibilities to assign the variables to terms. The rationale behind the second factor is assigning each of the max_vars variables a number in $\{1, \ldots, \texttt{max\_conj}\}$ that specifies to which term the variable belongs. Hence, it follows that

$$M \in \mathcal{O}(L \cdot \texttt{max\_vars}(\log(p) + \log(\texttt{max\_conj}))).$$

Since, by assumption, the parameters `max_vars` and `max_conj` both scale linearly with p and the parameter `nodesize` is constant, it follows with Theorem 2 that the computational complexity of logicDT is given by

$$\mathcal{O}\left(L \cdot n^2 p^2 \log(p)\right).$$

If it is assumed that the Markov chain length L is fixed, the computational complexity of logicDT becomes

$$\mathcal{O}\left(n^2 p^2 \log(p)\right).$$

The number of neighbor states per state in logicDT is in the magnitude of $\mathcal{O}(\texttt{max\_vars} \cdot p)$, since each variable in the state might be exchanged by another variable. Therefore, if instead the Markov chain length L is chosen in the magnitude of the number of neighbor states per state, the computational complexity of logicDT is given by

$$\mathcal{O}\left(n^2 p^4 \log(p)\right).$$

□

Appendix 4: Additional real data evaluations

In the following, logicDT, bagged logicDT, and the comparable methods are evaluated on 24 real data sets that were also analyzed in Aglin et al. (2020a) and Demirović et al. (2022). These data sets exclusively contain binary input variables and binary outcomes and were obtained from CP4IM[1] that provides (modified) data sets from the UCI Machine Learning Repository[2] that were modified by dichotimizing continuous variables into binary variables.

In Table 3, the dimensions of the considered data sets are summarized. Similar to Sect. 5, each method was applied to each data set 100 times using random splits into training, validation, and test data sets.

Figure 14 shows the predictive performance (as, again, measured by the AUC) of logicDT and the comparable methods in their applications to the 24 additional real data sets. This figure shows that logicDT achieves for most data sets a superior performance compared to conventional decision trees and DL8.5. logicDT seems to be on par with logic regression, since, in most cases, both methods yield similar results and, in the remaining cases, sometimes logicDT and sometimes logic regression induce better performances (see, e.g., the results from the applications to the vehicle and zoo-1 data set).

Ensemble methods that produce less interpretable models such as random forests, gradient boosting, and logic bagging yield better performances compared to logicDT for most data sets. However, when also considered logicDT in an ensemble framework, i.e., when considering bagged logicDT, then the performances are on a similar level as the other ensemble methods.

[1] CP4IM: https://dtai.cs.kuleuven.be/CP4IM/.

[2] UCI Machine Learning Repository: https://archive.ics.uci.edu.

Table 3 Dimensions of the 24 real data sets used for evaluating logicDT and the comparable methods

Data set	n	p	n_1	n_0
anneal	812	93	625	187
audiology	216	148	57	159
australian-credit	653	125	357	296
breast-wisconsin	683	120	444	239
diabetes	768	112	500	268
german-credit	1000	112	700	300
heart-cleveland	296	95	160	136
hepatitis	137	68	111	26
hypothyroid	3247	88	2970	277
ionosphere	351	445	225	126
kr-vs-kp	3196	73	1669	1527
letter	20,000	224	813	19,187
lymph	148	68	81	67
mushroom	8124	119	4208	3916
pendigits	7494	216	780	6714
primary-tumor	336	31	82	254
segment	2310	235	330	1980
soybean	630	50	92	538
splice-1	3190	287	1655	1535
tic-tac-toe	958	27	626	332
vehicle	846	252	218	628
vote	435	48	267	168
yeast	1484	89	463	1021
zoo-1	101	36	41	60

n denotes the sample size and p the number of input variables in the respective data set. n_1 and n_0 denote the numbers of observations with $Y = 1$ and $Y = 0$, respectively, since binary outcomes are considered

Appendix 5: Computation times

For the simulation study conducted in Sect. 5.1.1 and the application to the SALIA data conducted in Sect. 5.2, model fitting and prediction times were recorded. The calculations were performed using an Intel Xeon Gold 6346 CPU running on 3.6GHz. For the time measurement, no parallel computing was performed to reflect realistic single model evaluation times.

In Table 4, the mean model fitting and prediction times over ten replications is summarized. logicDT seems to be faster than logic regression, which also employs a stochastic search algorithm. In the application to the SALIA data, a more complex setting consisting of five terms was identified for logicDT compared to three terms for logic regression, which might explain the higher fitting time of logicDT compared to logic regression in the real data application.

Due to the computationally intensive global search, logicDT takes more time than comparable methods that employ greedy fitting algorithms such as conventional decision trees, random forests, gradient boosting, and MOB. Nonetheless, logicDT seems to

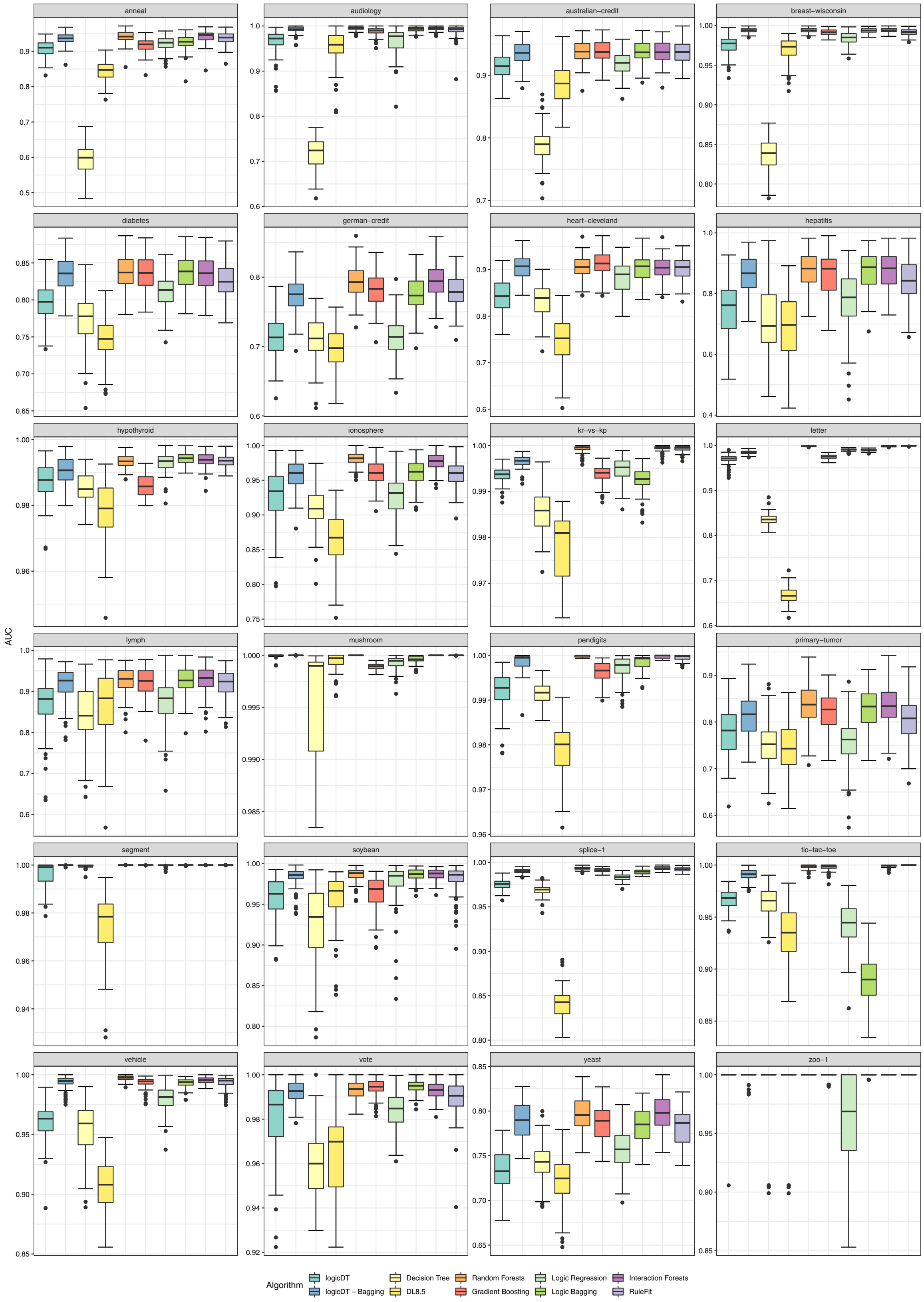

Fig. 14 Predictive performance of logicDT and the comparable methods in the evaluation of 24 real data sets

Table 4 Mean model evaluation times in seconds for the simulation study conducted in Sect. 5.1.1 and the real data application conducted in Sect. 5.2

Algorithm	Simulation scenario/study				
	Binary No E	Binary E	Continuous No E	Continuous E	SALIA
logicDT	29.334	87.615	12.826	33.414	38.727
logicDT–Bagging	260.063	307.279	82.151	770.853	1960.524
Decision Tree	0.184	0.183	0.184	0.179	0.186
DL8.5	2.907	3.571	–	–	700.399
Random Forests	5.704	6.440	6.875	7.133	1.980
Gradient Boosting	3.012	2.901	3.440	3.559	2.434
Logic Regression	27.004	206.816	33.671	22.710	15.803
Logic Bagging	82.584	40.730	57.809	63.202	575.047
MOB	–	0.513	–	0.479	–
Interaction Forests	82.682	344.738	322.515	501.945	40.416
RuleFit	77.568	96.647	71.394	78.370	92.420

The first line of the simulation scenario name corresponds to the considered outcome type (binary or continuous) and the second line corresponds to whether a continuous environmental covariable was incorporated (no E or E)

be faster than RuleFit. DL8.5 was faster than logicDT in the simulation study. For the real data application, DL8.5 was substantially slower than logicDT.

Bagged logicDT models take more time to be evaluated than bagged logic regression models. This is, in particular, due to the fact that the hyperparameter optimization for logic bagging identified `ntrees` = 3 with `nleaves` = 3 as the best setting for the simulation scenario with a binary outcome and an environmental covariable, i.e., a linear model involving three predictors, while bagged logicDT fits trees of depth of up to three in every greedy search step. This explanation also holds true for the other three scenarios and the real data application, since the hyperparameter optimization also yielded simpler settings for logic bagging compared to bagged logicDT.

Interaction forests are similarly fast as bagged logicDT and logic bagging in the simulation study. In the application to the SALIA data, interaction forests are comparably fast, since the hyperparameter optimization yielded for the number of randomly drawn input variable pairs per split `npairs` = 4, which is smaller than in the considered simulation study scenarios.

Unsurprisingly, for most methods, the computation time increases when also considering a continuous (environmental) covariable in comparison to not including a continuous (environmental) covariable. For some methods, this trend does not seem to be true, for example for logic regression, since the mean computation decreases when additionally considering a continuous covariable for a continuous outcome. However, this phenomenon is presumably caused by the identified hyperparameter setting, which is `ntrees` = 4 with `nleaves` = 8 for the continuous outcome scenario without a continuous covariable, corresponding to a rather complex model, and `ntrees` = 2 with `nleaves` = 3 for the continuous outcome scenario including a continuous covariable, corresponding to a rather simple model.

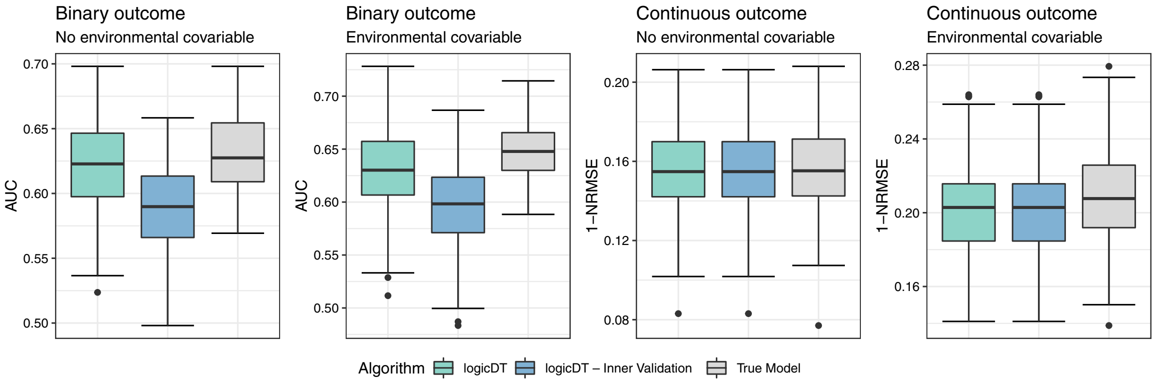

Fig. 15 Predictive performances of logicDT, logicDT using inner validation, and the true underlying model in the simulation study considering four different scenarios. The performance for binary outcomes is measured by the AUC and the performance for continuous outcomes is measured by the complement of the NRMSE (normalized root mean squared error)

Appendix 6: Inner validation

An idea to further robustify logicDT against overfitting might be to separate the decision tree fitting and evaluation steps in the search procedure by splitting the available training data into independent data sets for these two steps. We refer to this approach as *inner validation*, due to validating the states on independent validation data and the fitting procedure being nested in an outer validation that evaluates the performance of resulting logicDT models for tuning hyperparameters (see Sect. 3.6). This approach is similar to a nested cross-validation, which is, however, typically used for estimating unbiased prediction errors (see, e.g., Varma & Simon, 2006).

The trained logicDT model should not be heavily depending on the data split used such that a k-fold cross-validation approach is employed that randomly splits the training data into k approximately equally sized data sets $\mathcal{D}_1, \dots, \mathcal{D}_k$. For every $j \in \{1, \dots, k\}$, $k-1$ of these data sets $\mathcal{D}_{j'}$ ($j' \in \{1, \dots, k\} \setminus j$) are combined to one data set and used for training the decision trees (Line 9 in Algorithm 2) and the remaining data set $\mathcal{D}_j$ is used for computing the score (Line 10 in Algorithm 2). The total score of the state used to guide the search is then obtained by averaging the k scores.

In Fig. 15, the predictive performances of logicDT are summarized that were obtained using the aforementioned inner validation approach with 5-fold cross-validation in the simulation study presented in Sect. 5.1.1. For the binary outcome scenarios, the performance is worse compared to standard logicDT. For the continuous outcome scenarios, the performance is identical.

The performance loss can presumably be explained by the need to further split the available training data so that both the tree training step and the score computation step have to use less observations as opposed to standard logicDT. Moreover, the inner validation also leads to an increased computational burden due to fitting k trees in comparison to fitting a single tree in each search iteration. Therefore, the outer validation for hyperparameter optimization seems to be sufficient to balance the amount of underfitting and overfitting.

Acknowledgements The authors would like to thank Gaël Aglin for insightful discussions on the proper application of DL8.5. Computational infrastructure and support were provided by the Centre for Information and Media Technology at the Heinrich Heine University Düsseldorf.

Author Contributions ML and HS developed logicDT and the interaction VIM and designed the simulation study. ML and TS conceived the analyses of the real data application. The simulation study and the real data evaluations were conducted by ML. ML was the major contributor in writing the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. This work has been supported by the Research Training Group "Biostatistical Methods for High-Dimensional Data in Toxicology" (RTG 2624, Project R3) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation—Project Number 427806116). The SALIA follow-up study 2007–2010 was funded by the German Statutory Accident Insurance (DGUV) Grant No: 617.0-FP266 and the European Community's Seventh Framework Programme (FP 7/2007-2011) under the Grant Agreement Number 211250.

Data availability The simulated data sets are available upon request. The modified real data sets from the UCI Machine Learning Repository (https://archive.ics.uci.edu) that are analyzed in Appendix 4 have been downloaded in May 2023 from https://github.com/aia-uclouvain/pydl8.5.

Code availability The proposed methods are implemented and publicly available in the R package logicDT on CRAN (Lau, 2023).

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Consent for publication Not applicable.

Ethics approval and consent to participate The SALIA cohort study was conducted in accordance to the declaration of Helsinki and has been approved by the Ethics Committees of the Ruhr-University Bochum and the Heinrich Heine University Düsseldorf. We received written informed consent from all participants.

References

Aarts, E., & Van Laarhoven, P. (1985). Statistical cooling: A general approach to combinatorial optimization problems. *Philips Journal of Research, 40*(4), 193–226.

Aglin, G., Nijssen, S., & Schaus, P. (2020). Learning optimal decision trees using caching branch-and-bound search. In *Proceedings of the AAAI conference on artificial intelligence,* (Vol. 34, pp. 3146–3153).

Aglin, G., Nijssen, S., & Schaus, P. (2020b). PyDL8.5: A library for learning optimal decision trees. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI-20* (pp. 5222–5224). International Joint Conferences on Artificial Intelligence Organization.

Bellinger, C., Mohomed Jabbar, M. S., Zaïane, O., & Osornio-Vargas, A. (2017). A systematic review of data mining and machine learning for air pollution epidemiology. *BMC Public Health, 17*, 907. https://doi.org/10.1186/s12889-017-4914-3

Bénard, C., Biau, G., da Veiga, S., & Scornet, E. (2021). Interpretable random forests via rule extraction. In *Proceedings of the 24th international conference on artificial intelligence and statistics*, Volume 130 of *Proceedings of machine learning research* (pp. 937–945). PMLR.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological), 57*(1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11 (pp. 2546–2554). Curran Associates Inc.

Bertsimas, D., & Dunn, J. (2017). Optimal classification trees. *Machine Learning, 106*, 1039–1082. https://doi.org/10.1007/s10994-017-5633-9

Blockeel, H., & De Raedt, L. (1998). Top-down induction of first-order logical decision trees. *Artificial Intelligence, 101*(1), 285–297. https://doi.org/10.1016/S0004-3702(98)00034-4

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140. https://doi.org/10.1007/BF00058655

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC Press.

Bureau, A., Dupuis, J., Falls, K., Lunetta, K. L., Hayward, B., Keith, T. P., & Van Eerdewegh, P. (2005). Identifying SNPs predictive of phenotype using random forests. *Genetic Epidemiology, 28*(2), 171–182. https://doi.org/10.1002/gepi.20041

Carrizosa, E., Molero-Río, C., & Romero Morales, D. (2021). Mathematical optimization in classification and regression trees. *TOP, 29*, 5–33. https://doi.org/10.1007/s11750-021-00594-1

Che, R., & Motsinger-Reif, A. (2013). Evaluation of genetic risk score models in the presence of interaction and linkage disequilibrium. *Frontiers in Genetics, 4*, 138. https://doi.org/10.3389/fgene.2013.00138

Chen, C. C., Schwender, H., Keith, J., Nunkesser, R., Mengersen, K., & Macrossan, P. (2011). Methods for identifying SNP interactions: A review on variations of logic regression, random forest and Bayesian logistic regression. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 8*(6), 1580–1591. https://doi.org/10.1109/TCBB.2011.46

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '16, New York, NY, USA (pp. 785–794). Association for Computing Machinery.

Clarke, A., & Vyse, T. J. (2009). Genetics of rheumatic disease. *Arthritis Research & Therapy, 11*(5), 248. https://doi.org/10.1186/ar2781

Demirović, E., Lukina, A., Hebrard, E., Chan, J., Bailey, J., Leckie, C., Ramamohanarao, K., & Stuckey, P. J. (2022). MurTree: optimal decision trees via dynamic programming and search. *Journal of Machine Learning Research, 23*(26), 1–47.

Dudbridge, F., & Newcombe, P. J. (2015). Accuracy of gene scores when pruning markers by linkage disequilibrium. *Human Heredity, 80*(4), 178–186. https://doi.org/10.1159/000446581

Fokkema, M. (2020). Fitting prediction rule ensembles with R package pre. *Journal of Statistical Software, 92*(12), 1–30. https://doi.org/10.18637/jss.v092.i12

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Friedman, J. H., & Popescu, B. E. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics, 2*(3), 916–954. https://doi.org/10.1214/07-AOAS148

Fujimoto, K., Kojadinovic, I., & Marichal, J. L. (2006). Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior, 55*(1), 72–99. https://doi.org/10.1016/j.geb.2005.03.002

Györfi, L., Kohler, M., Krzyżak, A., & Walk, H. (2002). *A distribution-free theory of nonparametric regression*. Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.

Ho, D. S. W., Schierding, W., Wake, M., Saffery, R., & O'Sullivan, J. (2019). Machine learning SNP based prediction for precision medicine. *Frontiers in Genetics*. https://doi.org/10.3389/fgene.2019.00267

Hornung, R. (2022). Diversity forests: Using split sampling to enable innovative complex split procedures in random forests. *SN Computer Science, 3*(1), 1–16. https://doi.org/10.1007/s42979-021-00920-1

Hornung, R., & Boulesteix, A. L. (2022). Interaction forests: Identifying and exploiting interpretable quantitative and qualitative interaction effects. *Computational Statistics & Data Analysis, 171*, 107460. https://doi.org/10.1016/j.csda.2022.107460

Huang, M., Romeo, F., & Sangiovanni-Vincentelli, A. (1986). An efficient general cooling schedule for simulated annealing. In *Proceedings of the IEEE international conference on computer-aided design*, Santa Clara, California, USA (pp. 381–384). IEEE Computer Society.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Kooperberg, C., & Ruczinski, I. (2022). LogicReg: Logic regression. R Package Version 1.6.5.

Krämer, U., Herder, C., Sugiri, D., Strassburger, K., Schikowski, T., Ranft, U., & Rathmann, W. (2010). Traffic-related air pollution and incident type 2 diabetes: Results from the salia cohort study. *Environmental Health Perspectives, 118*(9), 1273–1279. https://doi.org/10.1289/ehp.0901689

Van Laarhoven, P., & Aarts, E. (1987). *Simulated annealing: Theory and applications*. Springer.

Lau, M. (2023). logicDT: Identifying interactions between binary predictors. R Package Version 1.0.3.

Lau, M., Wigmann, C., Kress, S., Schikowski, T., & Schwender, H. (2022). Evaluation of tree-based statistical learning methods for constructing genetic risk scores. *BMC Bioinformatics, 23*, 97. https://doi.org/10.1186/s12859-022-04634-w

Li, R. H., & Belford, G. G. (2002). Instability of decision tree classification algorithms. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining*, New York, NY, USA (pp. 570–575). Association for Computing Machinery.

Louppe, G. (2014). Understanding random forests: From theory to practice. Dissertation, University of Liège, Department of Electrical Engineering & Computer Science. arXiv:1407.7502.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S. I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence, 2*, 56–67. https://doi.org/10.1038/s42256-019-0138-9

Malley, J. D., Kruppa, J., Dasgupta, A., Malley, K. G., & Ziegler, A. (2012). Probability machines: Consistent probability estimation using nonparametric learning machines. *Methods of Information in Medicine, 51*(1), 74–81. https://doi.org/10.3414/ME00-01-0052

Meinshausen, N. (2010). Node harvest. *The Annals of Applied Statistics, 4*(4), 2049–2072. https://doi.org/10.1214/10-AOAS367

Mentch, L., & Hooker, G. (2016). Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *Journal of Machine Learning Research, 17*(26), 1–41.

Menze, B. H., Kelm, B. M., Splitthoff, D. N., Koethe, U., & Hamprecht, F. A. (2011). On oblique random forests. In *Proceedings of the joint European conference on machine learning and knowledge discovery in databases*, Berlin, Heidelberg (pp. 453–469). Springer.

Murthy, S. K., Kasif, S., & Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research, 2*, 1–32. https://doi.org/10.1613/jair.63

Murthy, S. K., & Salzberg, S. (1995). Decision tree induction: How effective is the greedy heuristic? In *Proceedings of the first international conference on knowledge discovery and data mining*, KDD'95 (pp. 222–227). AAAI Press.

Nijssen, S., & Fromont, E. (2010). Optimal constraint-based decision tree induction from itemset lattices. *Data Mining and Knowledge Discovery, 21*, 9–51. https://doi.org/10.1007/s10618-010-0174-x

Ottman, R. (1996). Gene-environment interaction: Definitions and study design. *Preventive Medicine, 25*(6), 764–770. https://doi.org/10.1006/pmed.1996.0117

Provost, F., & Domingos, P. (2003). Tree Induction for probability-based ranking. *Machine Learning, 52*(3), 199–215. https://doi.org/10.1023/A:1024099825458

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., De Bakker, P. I., Daly, M. J., & Pak, C. S. (2007). PLINK: A tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics, 81*(3), 559–575. https://doi.org/10.1086/519795

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc.

R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.

Ruczinski, I., Kooperberg, C., & LeBlanc, M. (2003). Logic regression. *Journal of Computational and Graphical Statistics, 12*(3), 475–511. https://doi.org/10.1198/1061860032238

Ruczinski, I., Kooperberg, C., & LeBlanc, M. (2004). Exploring interactions in high-dimensional genomic data: An overview of logic regression, with applications. *Journal of Multivariate Analysis, 90*(1), 178–195. https://doi.org/10.1016/j.jmva.2004.02.010

Rusch, T., & Zeileis, A. (2013). Gaining insight with recursive partitioning of generalized linear models. *Journal of Statistical Computation and Simulation, 83*(7), 1301–1315. https://doi.org/10.1080/00949655.2012.658804

Schikowski, T., Sugiri, D., Ranft, U., Gehring, U., Heinrich, J., Wichmann, H. E., & Krämer, U. (2005). Long-term air pollution exposure and living close to busy roads are associated with COPD in women. *Respiratory Research, 6*, 152. https://doi.org/10.1186/1465-9921-6-152

Schwender, H., & Ickstadt, K. (2007). Identification of SNP interactions using logic regression. *Biostatistics, 9*(1), 187–198. https://doi.org/10.1093/biostatistics/kxm024

Scornet, E., Biau, G., & Vert, J. P. (2015). Consistency of random forests. *The Annals of Statistics, 43*(4), 1716–1741. https://doi.org/10.1214/15-aos1321

So, H. C., & Sham, P. C. (2017). Improving polygenic risk prediction from summary statistics by an empirical Bayes approach. *Scientific Reports, 7*, 41262. https://doi.org/10.1038/srep41262

Sorokina, D., Caruana, R., Riedewald, M., & Fink, D. (2008). Detecting statistical interactions with additive groves of trees. In *Proceedings of the 25th international conference on machine learning*, ICML '08, New York, NY, USA (pp. 1000–1007). Association for Computing Machinery.

Tang, C., Garreau, D., & von Luxburg, U. (2018). When do random forests fail? In *Proceedings of the 32nd international conference on neural information processing systems*, NIPS'18, Montréal, Canada (pp. 2987–2997).

Therneau, T., & Atkinson, B. (2019). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-15.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Tomita, T. M., Browne, J., Shen, C., Chung, J., Patsolic, J. L., Falk, B., Priebe, C. E., Yim, J., Burns, R., Maggioni, M., & Vogelstein, J. T. (2020). Sparse projection oblique randomer forests. *Journal of Machine Learning Research, 21*(104), 1–39.

Triki, E., Collette, Y., & Siarry, P. (2005). A theoretical study on the behavior of simulated annealing leading to a new cooling schedule. *European Journal of Operational Research, 166*(1), 77–92. https://doi.org/10.1016/j.ejor.2004.03.035

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley-Interscience.

Vapnik, V. N. (2000). *The nature of statistical learning theory*. Springer.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics, 7*, 91. https://doi.org/10.1186/1471-2105-7-91

Watson, D. S., & Wright, M. N. (2021). Testing conditional independence in supervised learning algorithms. *Machine Learning, 110*, 2107–2129. https://doi.org/10.1007/s10994-021-06030-6

Wilks, S. S. (1938). The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics, 9*(1), 60–62. https://doi.org/10.1214/aoms/1177732360

Wilson, S. (2021). ParBayesianOptimization: Parallel Bayesian optimization of hyperparameters. R Package Version 1.2.4.

Winham, S. J., Colby, C. L., Freimuth, R. R., Wang, X., de Andrade, M., Huebner, M., & Biernacka, J. M. (2012). SNP interaction detection with random forests in high-dimensional genetic data. *BMC Bioinformatics, 13*, 164. https://doi.org/10.1186/1471-2105-13-164

Wright, M. N., & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software, 77*(1), 1–17. https://doi.org/10.18637/jss.v077.i01

Wright, M. N., Ziegler, A., & König, I. R. (2016). Do little interactions get lost in dark random forests? *BMC Bioinformatics, 17*, 145. https://doi.org/10.1186/s12859-016-0995-8

Yang, B. B., Shen, S. Q., & Gao, W. (2019). Weighted oblique decision trees. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, pp. 5621–5627).

Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics, 17*(2), 492–514. https://doi.org/10.1198/106186008X319331

Zhi, S., Li, Q., Yasui, Y., Edge, T., Topp, E., & Neumann, N. F. (2015). Assessing host-specificity of *Escherichia coli* using a supervised learning logic-regression-based analysis of single nucleotide polymorphisms in intergenic regions. *Molecular Phylogenetics and Evolution, 92*, 72–81. https://doi.org/10.1016/j.ympev.2015.06.007

Zhu, H., Murali, P., Phan, D., Nguyen, L., & Kalagnanam, J. (2020). A scalable MIP-based method for learning optimal multivariate decision trees. In *Advances in neural information processing systems* (Vol. 33, pp. 1771–1781). Curran Associates, Inc.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Chapter 5

Discussion

5.1 Summary

In this dissertation, tree-based statistical learning methods have been investigated and developed for constructing GRS with a focus on identifying and modeling gene–gene and GxE interaction effects.

First, it was evaluated in which situations and how the tree-based methods random forests and logic regression can be used in place of the standard (internal) GRS construction procedure employing the elastic net. As could be seen in Chapter 2, random forests with probability estimation trees and ensemble logic regression (with bagging) achieved in nearly all scenarios—including scenarios in which no interaction effects are present—superior GRS regarding the association with the outcome compared to the elastic net. This pattern could not only be observed in simulation studies but was also confirmed in a real data application to data from the SALIA study. Hence, it can be concluded that random forests or logic regression with bagging can be generally used as a substitute to the elastic net for constructing GRS if the predictive performance shall be maximized.

Next, based on the observation that common GRS-based GxE interaction tests lose statistical power by dividing the available data set into two disjoint sub data sets, a novel GxE interaction test has been developed that utilizes bagging and OOB predictions to avoid these data splits, and therefore, uses the complete data set for both constructing a GRS model and statistically testing the presence of a GxE interaction effect. Moreover, it was proposed to employ random forests as the GRS modeling procedure, as random forests can incorporate gene–gene interaction effects, induced relatively strongly predictive GRS models in Chapter 2, and already perform relatively well with standard hyperparameter settings [Probst et al., 2019]. The proposed GxE interaction test as well as established procedures were evaluated in simulation studies and a real data set from the SALIA study. As Chapter 3 showed, both bagging-based GxE interaction tests, that either employ random forests or elastic net for GRS construction, are valid statistical testing procedures and are able to induce a high statistical power in most scenarios. Therefore, it can be concluded that the newly proposed bagging-based GxE interaction tests should be employed in place of the standard GRS-based test. Especially if it is suspected that gene–gene interaction effects are involved, the random-forests-based test should be used that might be directly applied without extensive hyperparameter tuning.

Finally, going back to the problem of constructing GRS, the downside of the well-performing methods random forests and logic regression with bagging is the lack of interpretability, i.e., understanding the fitted GRS model and how predictions are made. Hence, logicDT, a statistical learning method that constructs a single decision tree and aims for high predictive performance while maintaining interpretability, has been developed. In contrast to standard decision trees, logicDT may split on predictors or Boolean conjunctions of predictors and performs a global stochastic search for the optimal set of splitting variables. By allowing splits on conjunctions of predictors, logicDT detects and reveals gene–gene interaction effects. Moreover, leaf regression models can be fitted for also detecting and properly modeling GxE interaction effects using an appropriate split detection mechanism that takes this advanced modeling into account. As could be seen in Chapter 4, it was proven that logicDT is a strongly consistent statistical learning procedure, i.e., logicDT asymptotically identifies the true regression function $\mu(\boldsymbol{x}) = \mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]$. Furthermore, logicDT induced high predictive performances in comparison to other interpretability-focused procedures in simulation studies, an application to data from the SALIA study, and additional real data sets from various application fields. Therefore, logicDT is able to fulfill its objectives by fitting highly predictive and interpretable classification or regression models that might be also used in application contexts other than GRS construction.

As could be also seen in Chapter 4, ensemble logicDT (with bagging) can achieve even higher predictive performances if the true underlying model is complex and consists of many influential predictors. Hence, to obtain an interpretable machine learning model from this highly predictive ensemble and to quantify specific genetic effects, an interaction VIM has been also proposed that estimates the importances of single predictors and interaction effects between predictors. This interaction VIM first captures the importance of general interactions between predictors (i.e., if a considered set of predictors interacts in any way considering the outcome) and identifies the most plausible Boolean conjunction responsible for this interaction effect. In principle, the interaction VIM could be used in conjunction with any supervised statistical learning method and is, thus, not limited to applications to logicDT models. Moreover, for the considered case of binary predictors, the logic VIM has been proposed that can be computed particularly fast. The empirical experiments conducted in Chapter 4 also showed that the interaction VIM yields in conjunction with the logic VIM reasonable estimates for the importances of predictors.

5.2 Outlook

This work focused on the analysis of SNP selections from genes, pathways, or prior studies resulting in mid-dimensional problems with not more than a few hundred SNPs. However, it might be also interesting to model larger SNP selections, e.g., all genotyped SNPs. This requires computationally efficient methods which becomes a particularly complex problem if flexible modeling by incorporating interaction effects is desired. As was shown in Chapter 4, the computational complexity of logicDT with simulated annealing scales polynomially with the number of predictors. An alternative idea might be to employ gradient boosting with greedily fitted logicDT models as base learner. Restricting the tree depth to one would lead to fitting tree stumps that potentially split on interaction terms. Such an ensemble of simple logicDT models could be interpreted as a linear interaction model in which effect sizes and interaction effects could be directly read off. Hence, in future work, this type of ensemble logicDT model could be investigated for its applicability to high-dimensional problems.

In Chapter 3, a novel GxE interaction testing procedure was presented. In Chapter 4, logicDT was proposed that can measure the influence of gene–gene interaction effects and model GxE interaction effects. A possible direction of future research might be to implement statistical testing into logicDT so that it can be statistically tested if certain SNPs affect the considered outcome or if gene–gene or GxE interaction effects are present. For assessing marginal SNP effects and gene–gene interaction effects, an idea might be to estimate the null distribution of variable importances for unimportant predictors/terms. This could be realized, e.g., using random permutations as done by Kursa and Rudnicki [2010] or employing knockoffs [Candès et al., 2018] as proposed by Watson and Wright [2021]. For statistically testing the presence of a GxE interaction, it could be utilized that a GxE interaction is exactly present if the leaf regression models in logicDT differ more than by their offsets (which can be interpreted as genetic main effects). For example, a likelihood-ratio test could be carried out for testing a difference in slope of the environmental variable for different genotypes.

The tree structure of logicDT leads to identifying different phenotype risk models for different genotypes. These phenotype risk models in the leaves may be constant or describe influences of non-genetic variables such as environmental exposures or lifestyle indicators. By including potential confounders as additional covariates in the leaf regression models, estimated effects of environmental risk factors can be adjusted for confounding. Moreover, as the effects of SNPs might be confounded as well, e.g., due to population structure [Yashin et al., 2016], the inclusion of covariates that confound genetic effects could lead to fitting sparser decision trees by discarding non-causal splits that are only deemed important if no information about the confounders is included. However, in practice, data of confounders might not be available or confounders might be completely unknown. In this case, one idea might be to consider GLMMs as leaf regression models, where random effects account for unknown confounders [Listgarten et al., 2010, Sul et al., 2018]. Hence, logicDT might be investigated and potentially extended for the ability to control for (known or unknown) confounding effects and to produce causal models in future research.

All considered analyses put an emphasis on binary (e.g., disease statuses) or continuous (e.g., quantitative biomarkers) outcomes. In future work, it could be investigated whether the tree-based methods random forests and logic regression could be also used for constructing GRS for other outcome types such as longevity/survival GRS [Timmers et al., 2019, Tesi et al., 2020] or GRS for multivariate/correlated outcomes [Bahda et al., 2023]. For both random forests and logic regression, there are versions for survival analysis [Ruczinski et al., 2003, Ishwaran et al., 2008, Tietz et al., 2019]. However, only for random forests, there exist so far extensions to multivariate outcomes [see, e.g., Segal and Xiao, 2011]. Moreover, logicDT could be extended to other outcome types similarly to random forests extensions.

Recently, risk scores have not only been constructed for the genome but also for other omics types including the transcriptome considering RNA-(ribonucleic acid)-based risk scores [Alaterre et al., 2021], the proteome considering individual protein levels [Ganz et al., 2016], and the epigenome considering individual DNA methylation states [Hüls and Czamara, 2020]. In contrast to GRS, the predictors of these alternative risk scores are usually measured on a continuous scale. Thus, logic regression could no longer be applied to construct these alternative risk scores. logicDT could be generalized to also consider continuous predictors for splitting the decision tree by generalizing the interaction notion from Boolean conjunctions to products of predictors or by considering conjunctions of (binary) decision rules such as $(X_i < a) \wedge (X_j \geq b)$. Hence, logicDT could be also potentially used for constructing the discussed alternative risk scores, possibly integrating multiple risk factor types into one interpretable model.

For linear regression, estimating interaction effects between predictors at a high statistical power requires a substantially larger sample size than solely estimating marginal effects [Gelman et al., 2020]. The same holds true for decision trees, since they recursively partition the predictor space so that fewer observations fall into leaves of deeper trees/longer conjunction chains. Moreover, effect sizes of individual SNPs on the development of complex diseases are usually relatively small [Stringer et al., 2011]. Hence, for reliably detecting gene–gene or GxE interaction effects—especially when considering cohort studies and relatively uncommon diseases or uncommon genetic variants, large sample sizes are required. In this work, data sets with not more than 2000 observations were considered. The analyzed data sets from the SALIA cohort study had a maximum of 560 observations. In future studies, it could be investigated whether the conventional tree-based methods random forests and logic regression and the newly proposed tree-based GxE interaction test as well as logicDT also perform well compared to standard linear approaches when considering larger sample sizes. Moreover, the proposed tree-based methods might be applied to data from (much) larger studies such as the UK Biobank ($n \approx$ 500,000) [Sudlow et al., 2015] or the German National Cohort (NAKO; genetic data not yet available; $n \approx$ 200,000 planned) [German National Cohort (GNC) Consortium, 2014] to construct GRS or to detect gene–gene or GxE interaction effects in the development of considered phenotypes.

5.3 Conclusion

In conclusion, several contributions have been made to improve GRS construction and applicability in this dissertation. For maximizing the predictive ability of GRS, tree-based statistical learning methods should be employed. If interpretability is also crucial, logicDT can be used for obtaining a highly predictive and comprehensible GRS model that can also quantify specific genetic effects using the novel interaction VIM. GxE interaction testing becomes more efficient by employing bagging in GRS construction, and hence, utilizing all data for both modeling and statistical testing.

Appendix A

Software packages

A.1 GRSxE

In the following, the most important functions from the R software package GRSxE are presented using their manual pages. In the GRSxE package, we have implemented methods for testing GxE interaction effects—including the novel bagging-based test that was proposed in Chapter 3/Lau et al. [2023]. The GRSxE package is publicly available on CRAN at https://CRAN.R-project.org/package=GRSxE [Lau, 2023].

Package:	GRSxE
Title:	Testing Gene-Environment Interactions Through Genetic Risk Scores
Version:	1.0.1
Description:	Statistical testing procedures for detecting GxE (gene-environment) interactions. The main focus lies on GRSxE interaction tests that aim at detecting GxE interactions through GRS (genetic risk scores). Moreover, a novel testing procedure based on bagging and OOB (out-of-bag) predictions is implemented for incorporating all available observations at both GRS construction and GxE testing [Lau et al., 2023].
License:	MIT
Imports:	glmnet, ranger, stats, utils
Author:	Michael Lau [aut, cre] <https://orcid.org/0000-0002-5327-8351>
Maintainer:	Michael Lau <michael.lau@hhu.de>
Repository:	CRAN
Date/Publication:	2023-10-30 14:00:05 UTC

`GRSxE` *Testing gene-environment interactions*

Description

Fitting and evaluating GRS (genetic risk scores) for testing the presence of GxE (gene-environment) interactions.

Usage

```
GRSxE(
  X,
  y,
  E,
  C = NULL,
  test.type = "bagging",
  B = 500,
  replace = TRUE,
  subsample = ifelse(replace, 1, 0.632),
  test.ind = sample(nrow(X), floor(nrow(X)/2)),
  grs.type = "rf",
  grs.args = list()
)
```

Arguments

`X` Matrix or data frame of genetic variables such as SNPs usually coded as 0-1-2.

`y` Numeric vector of the outcome/phenotype. Binary outcomes such as a disease status should be coded as 0-1 (control-case).

`E` Numeric vector of the environmental exposure.

`C` Optional data frame containing potentially confounding variables to be adjusted for.

`test.type` Testing type. The standard setting is `"bagging"`, which employs its OOB (out-of-bag) prediction mechanism such that the full data can be used for both training the GRS and testing the GxE interaction. Alternatively, this can be set to `"holdout"`, which requires splitting the available data into a training data set and test data set. For that, `test.ind` needs to be set to the data indices used for testing.

`B` The number of bagging iterations if `test.type = "bagging"` is used. Also used as the number of trees grown in the random forest if `grs.type = "rf"` is set.

`replace` Should sampling with or without replacement be performed? Only used if `test.type = "bagging"` is set.

`subsample` Subsample fraction if `test.type = "bagging"` is used.

`test.ind` Vector of indices in the supplied data for testing the GxE interaction. Only used if `test.type = "holdout"` is set. The standard setting corresponds to a random 50:50 training-test split.

`grs.type` Type of GRS to be constructed. Either `"rf"` for a random forest or `"elnet"` for an elastic net.

`grs.args` Optional list of arguments passed to the GRS fitting procedure.

Details

The GRS is usually constructed through random forests for taking gene-gene interactions into account and using its OOB (out-of-bag) prediction mechanism. Alternatively, a classical GRS construction approach can be employed by fitting an elastic net. Bagging can also be applied to fit multiple elastic net models to also be able to perform OOB predictions.

The advantage of OOB predictions is that they allow the GRS model to be constructed on the full available data, while performing unbiased predictions also on the full available data. Thus, both the GRS construction and the GxE interaction testing can utilize all observations.

If desired, sampling can be performed without replacement in contrast to the classical bagging approach that utilizes bootstrap sampling.

Potentially confounding variables can also be supplied that will then be adjusted for in the GxE interaction testing.

This function uses a GLM (generalized linear model) for modelling the marginal genetic effect, marginal environmental effect, the GRSxE interaction effect, and potential confounding effects. The fitted GLM is returned, which can be, e.g., inspected via `summary(...)` to retrieve the Wald test p-values for the individual terms. The p-value corresponding to the `G:E` term is the p-value for testing the presence of a GRSxE interaction.

Value

An object of class `glm` is returned, in which `G:E` describes the GRSxE term.

References

- Lau, M., Kress, S., Schikowski, T. & Schwender, H. (2023). Efficient gene–environment interaction testing through bootstrap aggregating. Scientific Reports 13:937. doi:10.1038/s41598-023-28172-4
- Lau, M., Wigmann C., Kress S., Schikowski, T. & Schwender, H. (2022). Evaluation of tree-based statistical learning methods for constructing genetic risk scores. BMC Bioinformatics 23:97. doi:10.1186/s12859-022-04634-w
- Breiman, L. (1996). Bagging predictors. Machine Learning 24:123–140. doi:10.1007/BF00058655
- Breiman, L. (2001). Random Forests. Machine Learning 45:5–32. doi:10.1023/A:1010933404324
- Friedman J., Hastie T. & Tibshirani R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software 33(1):1–22. doi:10.18637/jss.v033.i01

Examples

```
# Generate toy data
set.seed(101299)
maf <- 0.25
n.snps <- 10
N <- 500
X <- matrix(sample(0:2, n.snps * N, replace = TRUE,
                   prob = c((1-maf)^2, 1-(1-maf)^2-maf^2, maf^2)),
            ncol = n.snps)
colnames(X) <- paste("SNP", 1:n.snps, sep="")
E <- rnorm(N, 20, 10)
E[E < 0] <- 0

# Generate outcome with a GxE interaction
y.GxE <- -0.75 + log(2) * (X[,"SNP1"] != 0) +
  log(4) * E/20 * (X[,"SNP2"] != 0 & X[,"SNP3"] == 0) +
  rnorm(N, 0, 2)
# Test for GxE interaction (Wald test for G:E)
summary(GRSxE(X, y.GxE, E))

# Generate outcome without a GxE interaction
y.no.GxE <- -0.75 + log(2) * (X[,"SNP1"] != 0) +
  log(4) * E/20 + log(4) * (X[,"SNP2"] != 0 & X[,"SNP3"] == 0) +
  rnorm(N, 0, 2)
# Test for GxE interaction (Wald test for G:E)
summary(GRSxE(X, y.no.GxE, E))
```

`GxE` *Testing individual gene-environment interactions*

Description

Function for testing univariate GxE interactions, e.g., using single SNPs or a GRS.

Usage

```
GxE(G, y, E, C = NULL)
```

Arguments

`G` Numeric vector of a genetic variable such as a GRS (genetic risk score) or a SNP coded as 0-1-2.

`y` Numeric vector of the outcome/phenotype. Binary outcomes such as a disease status should be coded as 0-1 (control-case).

`E`	Numeric vector of the environmental exposure.
`C`	Optional data frame containing potentially confounding variables to be adjusted for.

Details

This function uses a GLM (generalized linear model) for modelling the marginal genetic effect, marginal environmental effect, the GxE interaction effect, and potential confounding effects. The fitted GLM is returned, which can be, e.g., inspected via `summary(...)` to retrieve the Wald test p-values for the individual terms. The p-value corresponding to the `G:E` term is the p-value for testing the presence of a GxE interaction.

Value

An object of class `glm` is returned, in which `G:E` describes the GxE term.

References

- Lau, M., Kress, S., Schikowski, T. & Schwender, H. (2023). Efficient gene–environment interaction testing through bootstrap aggregating. Scientific Reports 13:937. doi:10.1038/s41598-023-28172-4

A.2 logicDT

In the following, the most important functions from the R software package `logicDT` are presented using their manual pages. In the `logicDT` package, we have implemented the statistical learning procedure logicDT and the novel interaction VIM that were both proposed in Chapter 4/Lau et al. [2024]. The `logicDT` package is publicly available on CRAN at `https://CRAN.R-project.org/package=logicDT` [Lau, 2024].

Package:	`logicDT`
Title:	Identifying Interactions Between Binary Predictors
Version:	1.0.4
Description:	A statistical learning method that tries to find the best set of predictors and interactions between predictors for modeling binary or quantitative response data in a decision tree. Several search algorithms and ensembling techniques are implemented allowing for finetuning the method to the specific problem. Interactions with quantitative covariables can be properly taken into account by fitting local regression models. Moreover, a variable importance measure for assessing marginal and interaction effects is provided. Implements the procedures proposed by Lau et al. [2024].
License:	MIT
Imports:	`glmnet`, `graphics`, `stats`, `utils`
Author:	Michael Lau [aut, cre] <`https://orcid.org/0000-0002-5327-8351`>
Maintainer:	Michael Lau <`michael.lau@hhu.de`>
Repository:	CRAN
Date/Publication:	2024-01-19 13:10:02 UTC

`calcAUC` *Fast computation of the AUC w.r.t. to the ROC*

Description

This function computes the area under the receiver operating characteristic curve.

Usage

```
calcAUC(preds, y, fast = TRUE, sorted = FALSE)
```

Arguments

`preds`	Numeric vector of predicted scores
`y`	True binary outcomes coded as 0 or 1. Must be an integer vector.
`fast`	Shall the computation be as fast as possible?
`sorted`	Are the predicted scores already sorted increasingly? If so, this can slightly speed up the computation.

Value

The AUC between 0 and 1

`calcNRMSE` *Calculate the NRMSE*

Description

Computation of the normalized root mean squared error.

Usage

```
calcNRMSE(preds, y, type = "sd")
```

Arguments

`preds`	Numeric vector of predictions
`y`	True outcomes
`type`	`"sd"` uses the standard deviation of `y` for normalization. `"range"` uses the whole span of `y`.

Value

The NRMSE

`cooling.schedule` *Define the cooling schedule for simulated annealing*

Description

This function should be used to configure a search with simulated annealing.

Usage

```
cooling.schedule(
  type = "adaptive",
  start_temp = 1,
  end_temp = -1,
  lambda = 0.01,
  total_iter = 2e+05,
  markov_iter = 1000,
  markov_leave_frac = 1,
  acc_type = "probabilistic",
  frozen_def = "acc",
  frozen_acc_frac = 0.01,
  frozen_markov_count = 5,
  frozen_markov_mode = "total",
  start_temp_steps = 10000,
  start_acc_ratio = 0.95,
  auto_start_temp = TRUE,
  remember_models = TRUE,
  print_iter = 1000
)
```

Arguments

`type` Type of cooling schedule. `"adaptive"` (default) or `"geometric"`

`start_temp` Start temperature on a log10 scale. Only used if `auto_start_temp = FALSE`.

`end_temp` End temperature on a log10 scale. Only used if `type = "geometric"`.

`lambda` Cooling parameter for the adaptive schedule. Values between 0.01 and 0.1 are recommended such that in total, several hundred thousand iterations are performed. Lower values lead to a more fine search with more iterations while higher values lead to a more coarse search with less total iterations.

`total_iter` Total number of iterations that should be performed. Only used for the geometric cooling schedule.

`markov_iter` Number of iterations for each Markov chain. The standard value does not need to be tuned, since the temperature steps and number of iterations per chain act complementary to each other, i.e., less iterations can be compensated by smaller temperature steps.

`markov_leave_frac`

Fraction of accepted moves leading to an early temperature reduction. This is primarily used at (too) high temperatures lowering the temperature if essentially a random walk is performed. E.g., a value of 0.5 together with `markov_iter = 1000` means that the chain will be left if $0.5 \cdot 1000 = 500$ states were accepted in a single chain.

`acc_type` Type of acceptance function. The standard `"probabilistic"` uses the conventional function $\exp((\text{Score}_{\text{old}} - \text{Score}_{\text{new}})/t)$ for calculating the acceptance probability. `"deterministic"` accepts the new state, if and only if $\text{Score}_{\text{new}} - \text{Score}_{\text{old}} < t$.

`frozen_def` How to define a frozen chain. `"acc"` means that if less than `frozen_acc_frac` · `markov_iter` states with different scores were accepted in a single chain, this chain is marked as frozen.

"sd" declares a chain as frozen if the corresponding score standard deviation is zero. Several frozen chains indicate that the search is finished.

frozen_acc_frac
: If frozen_def = "acc", this parameter determines the fraction of iterations that define a frozen chain.

frozen_markov_count
: Number of frozen chains that need to be observed for finishing the search.

frozen_markov_mode
: Do the frozen chains have to occur consecutively ("consecutive") or is the total number of frozen chains relevant ("total")?

start_temp_steps
: Number of iterations that should be used for estimating the ideal start temperature if auto_start_temp = TRUE is set.

start_acc_ratio
: Acceptance ratio that should be achieved with the automatically configured start temperature.

auto_start_temp
: Should the start temperature be configured automatically? TRUE or FALSE

remember_models
: Should already evaluated models be saved in a 2-dimensional hash table to prevent fitting the same trees multiple times?

print_iter
: Number of iterations after which a progress report shall be printed.

Details

type = "adapative" (default) automatically choses the temperature steps by using the standard deviation of the scores in a Markov chain together with the current temperature to evaluate if equilibrium is achieved. If the standard deviation is small or the temperature is high, equilibrium can be assumed leading to a strong temperature reduction. Otherwise, the temperature is only merely lowered. The parameter lambda is essential to control how fast the schedule will be executed and, thus, how many total iterations will be performed.

type = "geometric" is the conventional approach which requires more fine-tuning. Here, temperatures are uniformly lowered on a log10 scale. Thus, a start and an end temperature have to be supplied.

Value

An object of class `cooling.schedule` which is a list of all necessary cooling parameters.

`getDesignMatrix` *Design matrix for the set of conjunctions*

Description

Transform the original predictor matrix X into the conjunction design matrix which contains for each conjunction a corresponding column.

Usage

```
getDesignMatrix(X, disj)
```

Arguments

`X`	The original (binary) predictor matrix. This has to be of type `integer`.
`disj`	The conjunction matrix which can, e.g., be extracted from a fitted `logicDT` model via `$disj`.

Value

The transformed design matrix.

logicDT *Fitting logic decision trees*

Description

Main function for fitting logicDT models.

Usage

```
## Default S3 method:
logicDT(
  X,
  y,
  max_vars = 3,
  max_conj = 3,
  Z = NULL,
  search_algo = "sa",
  cooling_schedule = cooling.schedule(),
  scoring_rule = "auc",
  tree_control = tree.control(),
  gamma = 0,
  simplify = "vars",
  val_method = "none",
  val_frac = 0.5,
  val_reps = 10,
  allow_conj_removal = TRUE,
  conjsize = 1,
  randomize_greedy = FALSE,
  greedy_mod = TRUE,
  greedy_rem = FALSE,
  max_gen = 10000,
  gp_sigma = 0.15,
  gp_fs_interval = 1,
  ...
)
```

```
## S3 method for class 'formula'
logicDT(formula, data, ...)
```

Arguments

`X` Matrix or data frame of binary predictors coded as 0 or 1.

`y` Response vector. 0-1 coding for binary responses. Otherwise, a regression task is assumed.

`max_vars` Maximum number of predictors in the set of predictors. For the set $[X_1 \wedge X_2^c, X_1 \wedge X_3]$, this parameter is equal to 4.

`max_conj` Maximum number of conjunctions/input variables for the decision trees. For the set $[X_1 \wedge X_2^c, X_1 \wedge X_3]$, this parameter is equal to 2.

`Z` Optional matrix or data frame of quantitative/continuous covariables. Multiple covariables allowed for splitting the trees. If leaf regression models (such as four parameter logistic models) shall be fitted, only the first given covariable is used.

`search_algo` Search algorithm for guiding the global search. This can either be `"sa"` for simulated annealing, `"greedy"` for a greedy search or `"gp"` for genetic programming.

`cooling_schedule`
Cooling schedule parameters if simulated annealing is used. The required object should be created via the function `cooling.schedule`.

`scoring_rule`
Scoring rule for guiding the global search. This can either be `"auc"` for the area under the receiver operating characteristic curve (default for binary reponses), `"deviance"` for the deviance, `"nce"` for the normalized cross entropy or `"brier"` for the Brier score. For regression purposes, the MSE (mean squared error) is automatically chosen.

`tree_control`
Parameters controlling the fitting of decision trees. This should be configured via the function `tree.control`.

`gamma` Complexity penalty added to the score. If `gamma` > 0 is given, `gamma` $\cdot ||m||_0$ is added to the score with $||m||_0$ being the total number of variables contained in the current model m. The main purpose of this penalty is for fitting logicDT stumps in conjunction with boosting. For regular logicDT models or bagged logicDT models, instead, the model complexity parameters `max_vars` and `max_conj` should be tuned.

`simplify` Should the final fitted model be simplified? This means, that unnecessary terms as a whole (`"conj"`) will be removed if they cannot improve the score. `simplify = "vars"` additionally tries to prune individual conjunctions by removing unnecessary variables in those. `simplify = "none"` will not modify the final model.

`val_method` Inner validation method. `"rv"` leads to a repeated validation where `val_reps` times the original data set is divided into `val_frac` · 100% validation data and (1 − `val_frac`) · 100% training data. `"bootstrap"` draws bootstrap samples and uses the out-of-bag data as validation data. `"cv"` employs cross-validation with `val_reps` folds.

`val_frac` Only used if `val_method = "rv"`. See description of `val_method`.

`val_reps` Number of inner validation partitionings.

`allow_conj_removal`

Should it be allowed to remove complete terms/conjunctions in the search? If a model with the specified exact number of terms is desired, this should be set to `FALSE`. If extensive hyperparameter optimizations are feasible, `allow_conj_removal = FALSE` with a proper search over `max_vars` and `max_conj` is advised for fitting single models. For bagging or boosting with a greedy search, `allow_conj_removal = TRUE` together with a small number for `max_vars = max_conj` is recommended, e.g., 2 or 3.

`conjsize` The minimum of training samples that have to belong to a conjunction. This parameters prevents including unnecessarily complex conjunctions that rarely occur.

`randomize_greedy`

Should the greedy search be randomized by only considering $\sqrt{\text{Neighbour states}}$ neighbors at each iteration, similar to random forests. Speeds up the greedy search but can lead to inferior results.

greedy_mod Should modifications of conjunctions be considered in a greedy search? greedy_mod = FALSE speeds up the greedy search but can lead to inferior results.

greedy_rem Should the removal of conjunctions be considered in a greedy search? greedy_rem = FALSE speeds up the greedy search but can lead to inferior results.

max_gen Maximum number of generations for genetic programming.

gp_sigma Parameter σ for fitness sharing in genetic programming. Very small values (e.g., 0.001) are recommended leading to only penalizing models which yield the exact same score.

gp_fs_interval

Interval for fitness sharing in genetic programming. The fitness calculation can be computationally expensive if many models exist in one generation. gp_fs_interval = 10 leads to performing fitness sharing only every 10th generation.

... Arguments passed to logicDT.default

formula An object of type formula describing the model to be fitted.

data A data frame containing the data for the corresponding formula object. Must also contain quantitative covariables if they should be included as well.

Details

logicDT is a method for finding response-associated interactions between binary predictors. A global search for the best set of predictors and interactions between predictors is performed trying to find the global optimal decision trees. On the one hand, this can be seen as a variable selection. On the other hand, Boolean conjunctions between binary predictors can be identified as impactful which is particularly useful if the corresponding marginal effects are negligible due to the greedy fashion of choosing splits in decision trees.

Three search algorithms are implemented:

- Simulated annealing. An exhaustive stochastic optimization procedure. Recommended for single models (without [outer] bagging or boosting).

- Greedy search. A very fast search always looking for the best possible improvement. Recommended for ensemble models.
- Genetic programming. A more or less intensive search holding several competetive models at each generation. Niche method which is only recommended if multiple (simple) models do explain the variation in the response.

Furthermore, the option of a so-called "inner validation" is available. Here, the search is guided using several train-validation-splits and the average of the validation performance. This approach is computationally expensive but can lead to more robust single models.

For minimizing the computation time, two-dimensional hash tables are used saving evaluated models. This is irrelevant for the greedy search but can heavily improve the fitting times when employing a search with simulated annealing or genetic programming, especially when choosing an inner validation.

Value

An object of class `logicDT`. This is a list containing

`disj` A matrix of the identified set of predictors and conjunctions of predictors. Each row corresponds to one term. Each entry corresponds to the column index in `X`. Negative values indicate negations. Missing values mean that the term does not contain any more variables.

`real_disj` Human readable form of `disj`. Here, variable names are directly depicted.

`score` Score of the best model. Smaller values are prefered.

`pet` Decision tree fitted on the best set of input terms. This is a list containing the pointer to the `C` representation of the tree and `R` representations of the tree structure such as the splits and predictions.

`ensemble` List of decision trees. Only relevant if inner validation was used.

`total_iter` The total number of search iterations, i.e., tested configurations by fitting a tree (ensemble) and evaluating it.

`prevented_evals`
: The number of prevented tree fittings by using the two-dimensional hash table.

`...`
: Supplied parameters of the functional call to `logicDT`.

Saving and Loading

logicDT models can be saved and loaded using `save(...)` and `load(...)`. The internal `C` structures will not be saved but rebuilt from the `R` representations if necessary.

References

- Lau, M., Schikowski, T. & Schwender, H. (2024). logicDT: A procedure for identifying response-associated interactions between binary predictors. Machine Learning 113(2):933–992. doi:10.1007/s10994-023-06488-6
- Breiman, L., Friedman, J., Stone, C. J. & Olshen, R. A. (1984). Classification and Regression Trees. CRC Press. doi:10.1201/9781315139470
- Kirkpatrick, S., Gelatt C. D. & Vecchi M. P. (1983). Optimization by Simulated Annealing. Science 220(4598):671–680. doi:10.1126/science.220.4598.671

Examples

```
# Generate toy data
set.seed(123)
maf <- 0.25
n.snps <- 50
N <- 2000
X <- matrix(sample(0:2, n.snps * N, replace = TRUE,
                   prob = c((1-maf)^2, 1-(1-maf)^2-maf^2, maf^2)),
            ncol = n.snps)
colnames(X) <- paste("SNP", 1:n.snps, sep="")
X <- splitSNPs(X)
Z <- matrix(rnorm(N, 20, 10), ncol = 1)
colnames(Z) <- "E"
Z[Z < 0] <- 0
y <- -0.75 + log(2) * (X[,"SNP1D"] != 0) +
  log(4) * Z/20 * (X[,"SNP2D"] != 0 & X[,"SNP3D"] == 0) +
  rnorm(N, 0, 1)


# Fit and evaluate single logicDT model
model <- logicDT(X[1:(N/2),], y[1:(N/2)],
                 Z = Z[1:(N/2),,drop=FALSE],
                 max_vars = 3, max_conj = 2,
                 search_algo = "sa",
                 tree_control = tree.control(
                   nodesize = floor(0.05 * nrow(X)/2)
                 ),
                 simplify = "vars",
                 allow_conj_removal = FALSE,
                 conjsize = floor(0.05 * nrow(X)/2))
calcNRMSE(predict(model, X[(N/2+1):N,],
                  Z = Z[(N/2+1):N,,drop=FALSE]), y[(N/2+1):N])
plot(model)
print(model)

# Fit and evaluate bagged logicDT model
model.bagged <- logicDT.bagging(X[1:(N/2),], y[1:(N/2)],
                                Z = Z[1:(N/2),,drop=FALSE],
                                bagging.iter = 50,
                                max_vars = 3, max_conj = 3,
                                search_algo = "greedy",
                                tree_control = tree.control(
                                  nodesize = floor(0.05 * nrow(X)/2)
                                ),
                                simplify = "vars",
                                conjsize = floor(0.05 * nrow(X)/2))
calcNRMSE(predict(model.bagged, X[(N/2+1):N,],
                  Z = Z[(N/2+1):N,,drop=FALSE]), y[(N/2+1):N])
print(model.bagged)

# Fit and evaluate boosted logicDT model
model.boosted <- logicDT.boosting(X[1:(N/2),], y[1:(N/2)],
                                  Z = Z[1:(N/2),,drop=FALSE],
                                  boosting.iter = 50,
                                  learning.rate = 0.01,
                                  subsample.frac = 0.75,
                                  replace = FALSE,
                                  max_vars = 3, max_conj = 3,
                                  search_algo = "greedy",
                                  tree_control = tree.control(
                                    nodesize = floor(0.05 * nrow(X)/2)
                                  ),
                                  simplify = "vars",
                                  conjsize = floor(0.05 * nrow(X)/2))
calcNRMSE(predict(model.boosted, X[(N/2+1):N,],
                  Z = Z[(N/2+1):N,,drop=FALSE]), y[(N/2+1):N])
print(model.boosted)

# Calculate VIMs (variable importance measures)
vims <- vim(model.bagged)
plot(vims)
print(vims)

# Single greedy model
model <- logicDT(X[1:(N/2),], y[1:(N/2)],
                 Z = Z[1:(N/2),,drop=FALSE],
                 max_vars = 3, max_conj = 2,
                 search_algo = "greedy",
                 tree_control = tree.control(
                   nodesize = floor(0.05 * nrow(X)/2)
                 ),
                 simplify = "vars",
                 allow_conj_removal = FALSE,
                 conjsize = floor(0.05 * nrow(X)/2))
calcNRMSE(predict(model, X[(N/2+1):N,],
                  Z = Z[(N/2+1):N,,drop=FALSE]), y[(N/2+1):N])
plot(model)
print(model)
```

`logicDT.bagging` *Fitting bagged logicDT models*

Description

Function for fitting bagged logicDT models.

Usage

```
## Default S3 method:
logicDT.bagging(X, y, Z = NULL, bagging.iter = 500, ...)

## S3 method for class 'formula'
logicDT.bagging(formula, data, ...)
```

Arguments

`X`	Matrix or data frame of binary predictors coded as 0 or 1.
`y`	Response vector. 0-1 coding for binary responses. Otherwise, a regression task is assumed.
`Z`	Optional matrix or data frame of quantitative/continuous covariables. Multiple covariables allowed for splitting the trees. If leaf regression models (such as four parameter logistic models) shall be fitted, only the first given covariable is used.
`bagging.iter`	Number of bagging iterations
`...`	Arguments passed to `logicDT`
`formula`	An object of type `formula` describing the model to be fitted.
`data`	A data frame containing the data for the corresponding `formula` object. Must also contain quantitative covariables if they should be included as well.

Details

Details on single logicDT models can be found in `logicDT`.

Value

An object of class `logic.bagged`. This is a list containing

`models`	A list of fitted `logicDT` models
`bags`	A list of observation indices which were used to train each model
`...`	Supplied parameters of the functional call to `logicDT.bagging`.

`plot.logicDT` *Plot a logic decision tree*

Description

This function plots a logicDT model on the active graphics device.

Usage

```
fancy.plot(x, cdot = FALSE, ...)

## S3 method for class 'logicDT'
plot(
  x,
  fancy = TRUE,
  x_scaler = 0.5,
  margin_scaler = 0.2,
  cex = 1,
  cdot = FALSE,
  ...
)
```

Arguments

`x`	An object of the class `logicDT`
`cdot`	Should a centered dot be used instead of a logical and for depicting interactions?
`...`	Arguments passed to fancy plotting function

`fancy` Should the fancy mode be used for plotting? Default is `TRUE`.

`x_scaler` Scaling factor on the horizontal axis for deeper trees, i.e., `x_scaler = 0.5` means that the horizontal distance between two adjacent nodes is halved for every vertical level.

`margin_scaler`
Margin factor. Smaller values lead to smaller margins.

`cex` Scaling factor for the plotted text elements.

Details

There are two plotting modes:

- `fancy = FALSE` which draws a tree with direct edges between the nodes. Leaves are represented by their prediction value which is obtained by the (observed) conditional mean.
- `fancy = TRUE` plots a tree similar to those in the `rpart` (Therneau and Atkinson, 2019) and `splinetree` (Neufeld and Heggeseth, 2019) `R` packages. The trees are drawn in an angular manner and if leaf regression models were fitted, appropriate plots of the fitted curves are depicted in the leaves. Otherwise, the usual prediction values are shown.

Value

No return value, called for side effects

References

- Therneau, T. & Atkinson, B. (2019). rpart: Recursive Partitioning and Regression Trees. `https://CRAN.R-project.org/package=rpart`
- Neufeld, A. & Heggeseth, B. (2019). splinetree: Longitudinal Regression Trees and Forests. `https://CRAN.R-project.org/package=splinetree`

`predict.logicDT` *Prediction for logicDT models*

Description

Supply new input data for predicting the outcome with a fitted logicDT model.

Usage

```
## S3 method for class 'logic.bagged'
predict(object, X, Z = NULL, type = "prob", ...)

## S3 method for class 'logic.boosted'
predict(object, X, Z = NULL, type = "prob", ...)

## S3 method for class 'logicDT'
predict(
  object,
  X,
  Z = NULL,
  type = "prob",
  ensemble = FALSE,
  leaves = "4pl",
  ...
)

## S3 method for class 'genetic.logicDT'
predict(
  object,
  X,
  Z = NULL,
  models = "best",
  n_models = 10,
  ensemble = NULL,
  leaves = "4pl",
  ...
)
```

Arguments

`object` Fitted `logicDT` model. Usually a product of a call to `logicDT`.

`X` Matrix or data frame of binary input data. This object should correspond to the binary matrix for fitting the model.

`Z` Optional quantitative covariables supplied as a matrix or data frame. Only used (and required) if the model was fitted using them.

`type` Prediction type. This can either be `"prob"` for probability estimates or `"class"` for (hard) classification of binary responses. Ignored for regression.

`...` Parameters supplied to `predict.logicDT`

`ensemble` If the model was fitted using the inner validation approach, shall the prediction be constructed using the final validated ensemble (`TRUE`) or using the single final tree (`FALSE`)?

`leaves` If leaf regression models (such as four parameter logistic models) were fitted, shall these models be used for the prediction (`"4pl"`) or shall the constant leaf means be used (`"constant"`)?

`models` Which logicDT models fitted via genetic programming shall be used for prediction? `"best"` leads to the single best model in the final generation, `"all"` uses the average over the final generation and `"n_models"` uses the `n_models` best models.

`n_models` How many models shall be used if `models = "n_models"` and genetic programming was employed?

Value

A numeric vector of predictions. For binary outcomes, this is a vector with estimates for $P(Y = 1 \mid X = x)$.

splitSNPs	*Split biallelic SNPs into binary variables*

Description

This function takes a matrix or data frame of SNPs coded as 0, 1, 2 or 1, 2, 3 and returns a data frame with twice as many columns. SNPs are splitted into dominant and recessive modes, i.e., for a SNP $\in \{0, 1, 2\}$, two variables $\text{SNP}_D = (\text{SNP} \neq 0)$ and $\text{SNP}_R = (\text{SNP} = 2)$ are generated.

Usage

```
splitSNPs(data)
```

Arguments

`data` A matrix or data frame only consisting of SNPs to be splitted

Value

A data frame of the splitted SNPs

tree.control	*Control parameters for fitting decision trees*

Description

Configure the fitting process of individual decision trees.

Usage

```
tree.control(
  nodesize = 10,
  split_criterion = "gini",
  alpha = 0.05,
  cp = 0.001,
  smoothing = "none",
  mtry = "none",
  covariable = "final_4pl"
)
```

Arguments

`nodesize` Minimum number of samples contained in a terminal node. This parameter ensures that enough samples are available for performing predictions which includes fitting regression models such as 4pL models.

`split_criterion` Splitting criterion for deciding when and how to split. The default is `"gini"`/`"mse"` which utilizes the Gini splitting criterion for binary risk estimation tasks and the mean squared error as impurity measure in regression tasks. Alternatively, `"4pl"` can be used if a quantitative covariable is supplied and the parameter `covariable` is chosen such that 4pL model fitting is enabled, i.e., `covariable = "final_4pl"` or `covariable = "full_4pl"`. A fast modeling alternative is given by `"linear"` which also requires the parameter `covariable` to be properly chosen, i.e., `covariable = "final_linear"` or `covariable = "full_linear"`.

`alpha` Significance threshold for the likelihood ratio tests when using `split_criterion = "4pl"` or `"linear"`. Only splits that achieve a p-value smaller than `alpha` are eligible.

`cp` Complexity parameter. This parameter determines by which amount the impurity has to be reduced to further split a node. Here, the total tree impurity is considered. See details for a specific formula. Only used if `split_criterion = "gini"` or `"mse"`.

`smoothing` Shall the leaf predictions for risk estimation be smoothed? `"laplace"` yields Laplace smoothing. The default is `"none"` which does not employ smoothing.

`mtry` Shall the tree fitting process be randomized as in random forests? Currently, only `"sqrt"` for using $\sqrt{p}$ random predictors at each node for splitting and "none" (default) for fitting conventional decision trees are supported.

`covariable` How shall optional quantitative covariables be handled? "constant" ignores them. Alternatively, they can be considered as splitting variables ("_split"), used for fitting 4pL models in each leaf ("_4pl"), or used for fitting linear models in each leaf ("_linear"). If either splitting or model fitting is chosen, one should state if this should be handled over the whole search ("full_", computationally expensive) or just the final trees ("final_"). Thus, "final_4pl" would lead to fitting 4pL models in each leaf but only for the final tree fitting.

Details

For the Gini or MSE splitting criterion, if any considered split s leads to

$$P(t) \cdot \Delta I(s,t) > \texttt{cp}$$

for a node t, the empirical node probability $P(t)$ and the impurity reduction $\Delta I(s,t)$, then the node is further splitted. If not, the node is declared as a leaf. For continuous outcomes, `cp` will be scaled by the empirical variance of `y` to ensure the right scaling, i.e., `cp <-cp * var(y)`. Since the impurity measure for continuous outcomes is the mean squared error, this can be interpreted as controlling the minimum reduction of the normalized mean squared error (NRMSE to the power of two).

If one chooses the 4pL or linear splitting criterion, likelihood ratio tests testing the alternative of better fitting individual models are employed. The corresponding test statistic asymptotically follows a χ^2 distribution where the degrees of freedom are given by the difference in the number of model parameters, i.e., leading to $2 \cdot 4 - 4 = 4$ degrees of freedom in the case of 4pL models and to $2 \cdot 2 - 2 = 2$ degrees of freedom in the case of linear models.

For binary outcomes, choosing to fit linear models for evaluating the splits or for modeling the leaves actually leads to fitting LDA (linear discriminant analysis) models.

Value

An object of class `tree.control` which is a list of all necessary tree parameters.

`vim` *Variable Importance Measures (VIMs)*

Description

Calculate variable importance measures (VIMs) based on different approaches.

Usage

```
vim(
  model,
  scoring_rule = "auc",
  vim_type = "logic",
  adjust = TRUE,
  interaction_order = 3,
  nodesize = NULL,
  alpha = 0.05,
  X_oob = NULL,
  y_oob = NULL,
  Z_oob = NULL,
  leaves = "4pl",
  ...
)
```

Arguments

`model` The fitted `logicDT` or `logic.bagged` model

`scoring_rule`

The scoring rule for assessing the model performance. As in `logicDT`, `"auc"`, `"nce"`, `"deviance"` and `"brier"` are possible for binary outcomes. For regression, the mean squared error is used.

`vim_type` The type of VIM to be calculated. This can either be `"logic"`, `"remove"` or `"permutation"`. See below for details.

`adjust` Shall adjusted interaction VIMs be additionally (to the VIMs of identified terms) computed? See below for details.

`interaction_order`
If `adjust = TRUE`, up to which interaction order shall adjusted interaction VIMs be computed?

`nodesize` If `adjust = TRUE`, how many observations need to be discriminated by an interaction in order to being considered? Similar to `conjsize` in `logicDT` and `nodesize` in `tree.control`.

`alpha` If `adjust = TRUE`, a further adjustment can be performed trying to identify the specific conjunctions responsible for the interaction of the considered binary predictors. `alpha` specifies the significance level for statistical tests testing the alternative of a difference in the response for specific conjunctions. `alpha = 0` leads to no further adjustment. See below for details.

`X_oob` The predictor data which should be used for calculating the VIMs. Preferably some type of validation data independent of the training data.

`y_oob` The outcome data for computing the VIMs. Preferably some type of validation data independent of the training data.

`Z_oob` The optional covariable data for computing the VIMs. Preferably some type of validation data independent of the training data.

`leaves` The prediction mode if regression models (such as 4pL models) were fitted in the leaves. As in `predict.logicDT`, `"4pl"` and `"constant"` are the possible settings.

`...` Parameters passed to the different VIM type functions. For `vim_type = "logic"`, the argument `average` can be specified as `"before"` or `"after"`. For `vim_type = "permutation"`, `n.perm` can be set to the number of random permutations. For `vim_type = "remove"`, `empty.model` can be specified as either `"none"` ignoring empty models with all predictive terms removed or `"mean"` using the response mean as prediction in the case of an empty model. See below for details.

Details

Three different VIM methods are implemented:

- Permutation VIMs: Random permutations of the respective identified logic terms
- Removal VIMs: Removing single logic terms
- Logic VIMs: Prediction with both possible outcomes of a logic term

Details on the calculation of these VIMs are given below.

By variable importance, importance of identified logic terms is meant. These terms can be single predictors or conjunctions between predictors in the spirit of this software package.

Value

A data frame with two columns:

`var` Short descriptions of the terms for which the importance was measured. For example `-X1^X2` for $X_1^c \wedge X_2$.

`vim` The actual calculated VIM values.

The rows of such a data frame are sorted decreasingly by the VIM values.

Permutation VIMs (Breiman & Cutler, 2003)

Permutation VIMs are computed by comparing the the model's performance using the original data and data with random permutations of single terms.

Removal VIMs

Removal VIMs are constructed by removing specific logic terms from the set of predictors, refitting the decision tree and comparing the performance to the original model. Thus, this approach requires that at least two terms were found by the algorithm. Therefore, no VIM will be calculated if `empty.model = "none"` was specified. Alternatively, `empty.model = "mean"` can be set to use the constant mean response model for approximating the empty model.

Logic VIMs (Lau et al., 2024)

Logic VIMs use the fact that Boolean conjunctions are Boolean variables themselves and therefore are equal to 0 or 1. To compute the VIM for a specific term, predictions are performed once for this term fixed to 0 and once for this term fixed to 1. Then, the arithmetic mean of these two (risk or regression) predictions is used for calculating the performance. This performance is then compared to the original one as in the other VIM approaches (`average = "before"`). Alternatively, predictions for each fixed 0-1 scenario of the considered term can be performed leading to individual performances which then are averaged and compared to the original performance (`average = "after"`).

Validation

Validation data sets which were not used in the fitting of the model are prefered preventing an overfitting of the VIMs themselves. These should be specified by the `_oob` arguments, if neither bagging nor inner validation was used for fitting the model.

Bagging

For the bagging version, out-of-bag (OOB) data are naturally used for the calculation of VIMs.

VIM Adjustment for Interactions (Lau et al., 2024)

Since decision trees can naturally include interactions between single predictors (especially when strong marginal effects are present as well), logicDT models might, e.g., include the single input variables X_1 and X_2 but not their interaction $X_1 \wedge X_2$ although an interaction effect is present. We, therefore, developed and implemented an adjustment approach for calculating VIMs for such unidentified interactions nonetheless. For predictors $X_{i_1}, \ldots, X_{i_k} =: Z$, this interaction importance is given by

$$\begin{aligned}\mathrm{VIM}(X_{i_1} \wedge \ldots \wedge X_{i_k}) = \mathrm{VIM}(X_{i_1}, \ldots, X_{i_k} \mid X \setminus Z) \\ - \sum_{\{j_1,\ldots,j_l\} \subsetneq \{i_1,\ldots,i_k\}} \mathrm{VIM}(X_{j_1} \wedge \ldots \wedge X_{j_l} \mid X \setminus Z)\end{aligned}$$

and can basically be applied to all black-box models. By $\mathrm{VIM}(A \mid X \setminus Z)$, the VIM of A considering the predictor set excluding the variables in Z is meant, i.e., the improvement of additionally considering A while regarding only the predictors in $X \setminus Z$. The proposed interaction VIM can be recursively calculated through

$$\mathrm{VIM}(X_{i_1} \wedge X_{i_2}) = \mathrm{VIM}(X_{i_1}, X_{i_2} \mid X \setminus Z) - \mathrm{VIM}(X_{i_1} \mid X \setminus Z) - \mathrm{VIM}(X_{i_2} \mid X \setminus Z)$$

for $Z = X_{i_1}, X_{i_2}$. This leads to the relationship

$$\mathrm{VIM}(X_{i_1} \wedge \ldots \wedge X_{i_k}) = \sum_{\{j_1,\ldots,j_l\} \subseteq \{i_1,\ldots,i_k\}} (-1)^{k-l} \cdot \mathrm{VIM}(X_{j_1}, \ldots, X_{j_l} \mid X \setminus Z).$$

Identification of Specific Conjunctions (Lau et al., 2024)

The aforementioned VIM adjustment approach only captures the importance of a general definition of interactions, i.e., it just considers the question whether some variables do interact in any way. Since logicDT is aimed at identifying specific conjunctions (and also assigns them VIMs if they were identified by `logicDT`), a further adjustment approach is implemented which tries to identify the specific conjunction leading to an interaction effect. The idea of this method is to consider the response for each possible scenario of the interacting variables, e.g., for $X_1 \wedge (X_2^c \wedge X_3)$ where the second term $X_2^c \wedge X_3$ was identified by `logicDT` and, thus, two interacting terms are regarded, the $2^2 = 4$ possible scenarios $\{(i,j) \mid i,j \in \{0,1\}\}$ are considered. For each setting, the corresponding response is compared with outcome values of the complementary set. For continuous outcomes, a two sample t-test (with Welch correction for potentially unequal variances) is performed comparing the means between these two groups. For binary outcomes, Fisher's exact test is performed testing different underlying case probabilities. If at least one test rejects the null hypothesis of equal outcomes (without adjusting for multiple testing), the combination with the lowest p-value is chosen as the explanatory term for the interaction effect. For example, if the most significant deviation results from $X_1 = 0$ and $(X_2^c \wedge X_3) = 1$ from the example above, the term $X_1^c \wedge (X_2^c \wedge X_3)$ is chosen.

References

- Lau, M., Schikowski, T. & Schwender, H. (2024). logicDT: A procedure for identifying response-associated interactions between binary predictors. Machine Learning 113(2):933–992. doi:10.1007/s10994-023-06488-6
- Breiman, L. (2001). Random Forests. Machine Learning 45:5–32. doi:10.1023/A:1010933404324
- Breiman, L. & Cutler, A. (2003). Manual on Setting Up, Using, and Understanding Random Forests V4.0. University of California, Berkeley, Department of Statistics. `https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf`

Bibliography

G. Aglin, S. Nijssen, and P. Schaus. Learning optimal decision trees using caching branch-and-bound search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3146–3153, 2020. doi:10.1609/aaai.v34i04.5711.

E. Alaterre, V. Vikova, A. Kassambara, A. Bruyer, N. Robert, G. Requirand, C. Bret, C. Herbaux, L. Vincent, G. Cartron, O. Elemento, and J. Moreaux. RNA-sequencing-based transcriptomic score with prognostic and theranostic values in multiple myeloma. *Journal of Personalized Medicine*, 11(10):988, 2021. doi:10.3390/jpm11100988.

K. G. Ardlie, L. Kruglyak, and M. Seielstad. Patterns of linkage disequilibrium in the human genome. *Nature Reviews Genetics*, 3(4):299–309, 2002. doi:10.1038/nrg777.

A. Badré, L. Zhang, W. Muchero, J. C. Reynolds, and C. Pan. Deep neural network improves the estimation of polygenic risk scores for breast cancer. *Journal of Human Genetics*, 66(4):359–369, 2021. doi:10.1038/s10038-020-00832-7.

M. Bahda, J. Ricard, S. L. Girard, M. Maziade, M. Isabelle, and A. Bureau. Multivariate extension of penalized regression on summary statistics to construct polygenic risk scores for correlated traits. *Human Genetics and Genomics Advances*, 4(3):100209, 2023. doi:10.1016/j.xhgg.2023.100209.

R. Beelen, O. Raaschou-Nielsen, M. Stafoggia, Z. J. Andersen, G. Weinmayr, B. Hoffmann, K. Wolf, E. Samoli, P. Fischer, M. Nieuwenhuijsen, P. Vineis, W. W. Xun, K. Katsouyanni, K. Dimakopoulou, A. Oudin, B. Forsberg, L. Modig, A. S. Havulinna, T. Lanki, A. Turunen, B. Oftedal, W. Nystad, P. Nafstad, U. De Faire, N. L. Pedersen, C.-G. Östenson, L. Fratiglioni, J. Penell, M. Korek, G. Pershagen, K. T. Eriksen, K. Overvad, T. Ellermann,

M. Eeftens, P. H. Peeters, K. Meliefste, M. Wang, B. B. de Mesquita, D. Sugiri, U. Krämer, J. Heinrich, K. de Hoogh, T. Key, A. Peters, R. Hampel, H. Concin, G. Nagel, A. Ineichen, E. Schaffner, N. Probst-Hensch, N. Künzli, C. Schindler, T. Schikowski, M. Adam, H. Phuleria, A. Vilier, F. Clavel-Chapelon, C. Declercq, S. Grioni, V. Krogh, M.-Y. Tsai, F. Ricceri, C. Sacerdote, C. Galassi, E. Migliore, A. Ranzi, G. Cesaroni, C. Badaloni, F. Forastiere, I. Tamayo, P. Amiano, M. Dorronsoro, M. Katsoulis, A. Trichopoulou, B. Brunekreef, and G. Hoek. Effects of long-term exposure to air pollution on natural-cause mortality: an analysis of 22 European cohorts within the multicentre ESCAPE project. *The Lancet*, 383(9919):785–795, 2014. doi:10.1016/S0140-6736(13)62158-3.

D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning*, 106: 1039–1082, 2017. doi:10.1007/s10994-017-5633-9.

S. Bordt and U. von Luxburg. From Shapley values to generalized additive models and back. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 709–745. PMLR, 2023. URL `https://proceedings.mlr.press/v206/bordt23a.html`.

V. Botta, G. Louppe, P. Geurts, and L. Wehenkel. Exploiting SNP correlations within random forest for genome-wide association studies. *PLOS ONE*, 9(4): 1–11, 2014. doi:10.1371/journal.pone.0093379.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. doi:10.1007/BF00058655.

L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001. doi:10.1023/A:1010933404324.

L. Breiman and A. Cutler. Manual on setting up, using, and understanding Random Forests v4.0. *University of California, Berkeley, Department of Statistics*, 2003. URL `https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf`.

L. Breiman, J. H. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees.* CRC Press, Boca Raton, Florida, USA, 1984. doi:10.1201/9781315139470.

A. Bureau, J. Dupuis, K. Falls, K. L. Lunetta, B. Hayward, T. P. Keith, and P. Van Eerdewegh. Identifying SNPs predictive of phenotype using random forests. *Genetic Epidemiology*, 28(2):171–182, 2005. doi:10.1002/gepi.20041.

M. P. Calus and J. Vandenplas. SNPrune: an efficient algorithm to prune large SNP array and sequence datasets based on high linkage disequilibrium. *Genetics Selection Evolution*, 50:34, 2018. doi:10.1186/s12711-018-0404-z.

E. Candès, Y. Fan, L. Janson, and J. Lv. Panning for gold: 'model-x' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(3):551–577, 2018. doi:10.1111/rssb.12265.

E. Carrizosa, C. Molero-Río, and D. Romero Morales. Mathematical optimization in classification and regression trees. *TOP*, 29:5–33, 2021. doi:10.1007/s11750-021-00594-1.

C. C. Chen, H. Schwender, J. Keith, R. Nunkesser, K. Mengersen, and P. Macrossan. Methods for identifying SNP interactions: A review on variations of logic regression, random forest and Bayesian logistic regression. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(6):1580–1591, 2011. doi:10.1109/TCBB.2011.46.

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2939672.2939785.

S. W. Choi, T. S.-H. Mak, and P. F. O'Reilly. Tutorial: a guide to performing polygenic risk score analyses. *Nature Protocols*, 15(9):2759–2772, 2020. doi:10.1038/s41596-020-0353-1.

S. Clancy. Genetic mutation. *Nature Education*, 1(1):187, 2008. URL `https://www.nature.com/scitable/topicpage/genetic-mutation-441/`.

E. Demirović, A. Lukina, E. Hebrard, J. Chan, J. Bailey, C. Leckie, K. Ramamohanarao, and P. J. Stuckey. MurTree: optimal decision trees via dynamic programming and search. *Journal of Machine Learning Research*, 23(26):1–47, 2022. URL `http://jmlr.org/papers/v23/20-520.html`.

I. Dinu, S. Mahasirimongkol, Q. Liu, H. Yanai, N. Sharaf Eldin, E. Kreiter, X. Wu, S. Jabbari, K. Tokunaga, and Y. Yasui. SNP-SNP interactions discovered by logic regression explain crohn's disease genetics. *PLOS ONE*, 7(10):1–6, 2012. doi:10.1371/journal.pone.0043035.

M. Eeftens, R. Beelen, K. de Hoogh, T. Bellander, G. Cesaroni, M. Cirach, C. Declercq, A. Dėdelė, E. Dons, A. de Nazelle, K. Dimakopoulou, K. Eriksen, G. Falq, P. Fischer, C. Galassi, R. Gražulevičienė, J. Heinrich, B. Hoffmann, M. Jerrett, D. Keidel, M. Korek, T. Lanki, S. Lindley, C. Madsen, A. Mölter, G. Nádor, M. Nieuwenhuijsen, M. Nonnemacher, X. Pedeli, O. Raaschou-Nielsen, E. Patelarou, U. Quass, A. Ranzi, C. Schindler, M. Stempfelet, E. Stephanou, D. Sugiri, M.-Y. Tsai, T. Yli-Tuomi, M. J. Varró, D. Vienneau, S. v. Klot, K. Wolf, B. Brunekreef, and G. Hoek. Development of land use regression models for PM2.5, PM2.5 absorbance, PM10 and PMcoarse in 20 European study areas; results of the ESCAPE project. *Environmental Science & Technology*, 46(20): 11195–11205, 2012. doi:10.1021/es301948k.

M. Eeftens, H. C. Phuleria, R. Meier, I. Aguilera, E. Corradi, M. Davey, R. Ducret-Stich, M. Fierz, R. Gehrig, A. Ineichen, D. Keidel, N. Probst-Hensch, M. S. Ragettli, C. Schindler, N. Künzli, and M.-Y. Tsai. Spatial and temporal variability of ultrafine particles, NO2, PM2.5, PM2.5 absorbance, PM10 and PMcoarse in Swiss study areas. *Atmospheric Environment*, 111:60–70, 2015. ISSN 1352-2310. doi:10.1016/j.atmosenv.2015.03.031.

I. C. Eze, L. G. Hemkens, H. C. Bucher, B. Hoffmann, C. Schindler, N. Künzli, T. Schikowski, and N. M. Probst-Hensch. Association between ambient air pollution and diabetes mellitus in europe and north america: Systematic review and meta-analysis. *Environmental Health Perspectives*, 123(5):381–389, 2015. doi:10.1289/ehp.1307823.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi:10.1214/aos/1013203451.

J. H. Friedman and B. E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916 – 954, 2008. doi:10.1214/07-AOAS148.

W. J. Fu. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998. doi:10.2307/1390712.

K. Fujimoto, I. Kojadinovic, and J.-L. Marichal. Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior*, 55(1):72–99, 2006. doi:10.1016/j.geb.2005.03.002.

P. Ganz, B. Heidecker, K. Hveem, C. Jonasson, S. Kato, M. R. Segal, D. G. Sterling, and S. A. Williams. Development and validation of a protein-based risk score for cardiovascular outcomes among patients with stable coronary heart disease. *JAMA*, 315(23):2532–2541, 2016. doi:10.1001/jama.2016.5951.

W. J. Gauderman, P. Zhang, J. L. Morrison, and J. P. Lewinger. Finding novel genes by testing G×E interactions in a genome-wide association study. *Genetic Epidemiology*, 37(6):603–613, 2013. doi:10.1002/gepi.21748.

W. J. Gauderman, B. Mukherjee, H. Aschard, L. Hsu, J. P. Lewinger, C. J. Patel, J. S. Witte, C. Amos, C. G. Tai, D. Conti, D. G. Torgerson, S. Lee, and N. Chatterjee. Update on the state of the science for analytical methods for gene-environment interactions. *American Journal of Epidemiology*, 186(7):762–770, 2017. doi:10.1093/aje/kwx228.

A. Gelman, J. Hill, and A. Vehtari. *Regression and Other Stories*. Cambridge University Press, Cambridge, UK, 2020. doi:10.1017/9781139161879.

C. George Priya Doss, C. Sudandiradoss, R. Rajasekaran, P. Choudhury, P. Sinha, P. Hota, U. P. Batra, and S. Rao. Applications of computational algorithm tools to identify functional SNPs. *Functional & Integrative Genomics*, 8:309–316, 2008. doi:10.1007/s10142-008-0086-7.

German National Cohort (GNC) Consortium. The German National Cohort: aims, study design and organization. *European Journal of Epidemiology*, 29(5):371–382, 2014. doi:10.1007/s10654-014-9890-7.

D. Gilbert-Diamond and J. H. Moore. Analysis of gene-gene interactions. *Current Protocols in Human Genetics*, 70(1):1.14.1–1.14.12, 2011. doi:10.1002/0471142905.hg0114s70.

D. Gola, J. Erdmann, B. Müller-Myhsok, H. Schunkert, and I. R. König. Polygenic risk scores outperform machine learning methods in predicting coronary artery disease status. *Genetic Epidemiology*, 44(2):125–138, 2020. doi:10.1002/gepi.22279.

J. Graw. *Genetik*. Springer-Verlag Berlin Heidelberg, 6th edition, 2015. doi:10.1007/978-3-662-44817-5.

L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer, New York, NY, USA, 2002. doi:10.1007/b97848.

S. S. Hada and M. A. Carreira-Perpiñán. Sparse oblique decision trees: A tool to interpret natural language processing datasets. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022. doi:10.1109/IJCNN55064.2022.9891903.

A. Hajat, R. F. MacLehose, A. Rosofsky, K. D. Walker, and J. E. Clougherty. Confounding by socioeconomic status in epidemiological studies of air pollution and health: Challenges and opportunities. *Environmental Health Perspectives*, 129(6):065001, 2021. doi:10.1289/EHP7980.

T. Hastie and R. Tibshirani. Varying-coefficient models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(4):757–796, 1993. doi:10.1111/j.2517-6161.1993.tb01939.x.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media, New York, NY, USA, 2009. doi:10.1007/978-0-387-84858-7.

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC, New York, NY, USA, 2015. doi:10.1201/b18401.

J. N. Hirschhorn, K. Lohmueller, E. Byrne, and K. Hirschhorn. A comprehensive review of genetic association studies. *Genetics in Medicine*, 4(2):45–61, 2002. doi:10.1097/00125817-200203000-00002.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi:10.1080/00401706.1970.10488634.

A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, and W. Samek. xxAI - beyond explainable artificial intelligence. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pages 3–10. Springer International Publishing, 2022. doi:10.1007/978-3-031-04083-2_1.

L. Hsu, S. Jiao, J. Y. Dai, C. Hutter, U. Peters, and C. Kooperberg. Powerful cocktail methods for detecting genome-wide gene-environment interaction. *Genetic Epidemiology*, 36(3):183–194, 2012. doi:10.1002/gepi.21610.

A. Hüls, K. Ickstadt, T. Schikowski, and U. Krämer. Detection of gene-environment interactions in the presence of linkage disequilibrium and noise by using genetic risk scores with internal weights from elastic net regression. *BMC Genetics*, 18:55, 2017a. doi:10.1186/s12863-017-0519-1.

A. Hüls, U. Krämer, C. Carlsten, T. Schikowski, K. Ickstadt, and H. Schwender. Comparison of weighting approaches for genetic risk scores in gene-environment interaction studies. *BMC Genetics*, 18:115, 2017b. doi:10.1186/s12863-017-0586-3.

A. Hüls and D. Czamara. Methodological challenges in constructing DNA methylation risk scores. *Epigenetics*, 15(1-2):1–11, 2020. doi:10.1080/15592294.2019.1644879.

International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001. doi:10.1038/35057062.

H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860, 2008. doi:10.1214/08-AOAS169.

S. Janitza, C. Strobl, and A.-L. Boulesteix. An AUC-based permutation variable importance measure for random forests. *BMC Bioinformatics*, 14:119, 2013. doi:10.1186/1471-2105-14-119.

S. Jiao, L. Hsu, S. Bézieau, H. Brenner, A. T. Chan, J. Chang-Claude, L. Le Marchand, M. Lemire, P. A. Newcomb, M. L. Slattery, and U. Peters. SBERIA: Set-based gene-environment interaction test for rare and common variants in complex diseases. *Genetic Epidemiology*, 37(5):452–464, 2013. doi:10.1002/gepi.21735.

R. Johnston, K. Jones, and D. Manley. Confounding and collinearity in regression analysis: a cautionary tale and an alternative procedure, illustrated by studies of British voting behaviour. *Quality & Quantity*, 52:1957–1976, 2018. doi:10.1007/s11135-017-0584-6.

M. Jokar and M. Jokar. Prevalence of inflammatory rheumatic diseases in a rheumatologic outpatient clinic: analysis of 12626 cases. *Rheumatology Research*, 3(1):21–27, 2018. doi:10.22631/rr.2017.69997.1037.

A. S. Kampstra and R. E. Toes. HLA class II and rheumatoid arthritis: the bumpy road of revelation. *Immunogenetics*, 69(8):597–603, 2017. doi:10.1007/s00251-017-0987-5.

P. Khankhanian, L. Din, S. J. Caillier, P.-A. Gourraud, and S. E. Baranzini. SNP imputation bias reduces effect size determination. *Frontiers in Genetics*, 6, 2015. doi:10.3389/fgene.2015.00030.

Y. Kirino and E. F. Remmers. Genetic architectures of seropositive and seronegative rheumatic diseases. *Nature Reviews Rheumatology*, 11(7):401–414, 2015. doi:10.1038/nrrheum.2015.41.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi:10.1126/science.220.4598.671.

C. Kooperberg and I. Ruczinski. Identifying interacting SNPs using Monte Carlo logic regression. *Genetic Epidemiology*, 28(2):157–170, 2005. doi:10.1002/gepi.20042.

C. Kooperberg and I. Ruczinski. *LogicReg: Logic Regression*, 2023. R package version 1.6.6.

J. Kruppa, A. Ziegler, and I. R. König. Risk estimation and risk prediction using machine-learning methods. *Human Genetics*, 131(10):1639–1654, 2012.

M. B. Kursa and W. R. Rudnicki. Feature selection with the Boruta package. *Journal of Statistical Software*, 36(11):1–13, 2010. doi:10.18637/jss.v036.i11.

M. Lau. Evaluation of tree-based classification and regression methods for constructing genetic risk scores. Master's thesis, Mathematical Institute, Heinrich Heine University, Düsseldorf, 2020.

M. Lau. *GRSxE: Testing Gene-Environment Interactions Through Genetic Risk Scores*, 2023. URL `https://CRAN.R-project.org/package=GRSxE`. R package version 1.0.1.

M. Lau. *logicDT: Identifying Interactions Between Binary Predictors*, 2024. URL `https://CRAN.R-project.org/package=logicDT`. R package version 1.0.4.

M. Lau, C. Wigmann, S. Kress, T. Schikowski, and H. Schwender. Evaluation of tree-based statistical learning methods for constructing genetic risk scores. *BMC Bioinformatics*, 23:97, 2022. doi:10.1186/s12859-022-04634-w.

M. Lau, S. Kress, T. Schikowski, and H. Schwender. Efficient gene–environment interaction testing through bootstrap aggregating. *Scientific Reports*, 13:937, 2023. doi:10.1038/s41598-023-28172-4.

M. Lau, T. Schikowski, and H. Schwender. logicDT: a procedure for identifying response-associated interactions between binary predictors. *Machine Learning*, 113(2):933–992, 2024. doi:10.1007/s10994-023-06488-6.

C. M. Lewis and E. Vassos. Prospects for using risk scores in polygenic medicine. *Genome Medicine*, 9:96, 2017. doi:10.1186/s13073-017-0489-y.

C. M. Lewis and E. Vassos. Polygenic risk scores: from research tools to clinical instruments. *Genome Medicine*, 12:44, 2020. doi:10.1186/s13073-020-00742-5.

R.-H. Li and G. G. Belford. Instability of decision tree classification algorithms. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 570–575, New York, NY, USA, 2002. Association for Computing Machinery. doi:10.1145/775047.775131.

M. Lim and T. Hastie. Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics*, 24(3):627–654, 2015. doi:10.1080/10618600.2014.938812.

W.-Y. Lin, C.-C. Huang, Y.-L. Liu, S.-J. Tsai, and P.-H. Kuo. Polygenic approaches to detect gene–environment interactions when external information is unavailable. *Briefings in Bioinformatics*, 20(6):2236–2252, 2019. doi:10.1093/bib/bby086.

X. Lin, S. Lee, D. C. Christiani, and X. Lin. Test for interactions between a genetic marker set and environment in generalized linear models. *Biostatistics*, 14(4): 667–681, 2013. doi:10.1093/biostatistics/kxt006.

X. Lin, S. Lee, M. C. Wu, C. Wang, H. Chen, Z. Li, and X. Lin. Test for rare variants by environment interactions in sequencing association studies. *Biometrics*, 72(1):156–164, 2016. doi:10.1111/biom.12368.

J. Listgarten, C. Kadie, E. E. Schadt, and D. Heckerman. Correction for hidden confounders in the genetic analysis of gene expression. *Proceedings of the National Academy of Sciences*, 107(38):16465–16470, 2010. doi:10.1073/pnas.1002425107.

M. Loecher. Debiasing MDI feature importance and SHAP values in tree ensembles. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 114–129. Springer International Publishing, 2022. doi:10.1007/978-3-031-14463-9_8.

M. Loecher. Debiasing SHAP scores in random forests. *AStA Advances in Statistical Analysis*, 2023. doi:10.1007/s10182-023-00479-7.

G. Louppe. *Understanding Random Forests: From Theory to Practice.* Dissertation, University of Liège, Department of Electrical Engineering & Computer Science, 2014.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2: 56–67, 2020. doi:10.1038/s42256-019-0138-9.

W. Ma, Y.-L. Lau, W. Yang, and Y.-F. Wang. Random forests algorithm boosts genetic risk prediction of systemic lupus erythematosus. *Frontiers in Genetics*, 13, 2022. doi:10.3389/fgene.2022.902793.

A. Majumdar, K. S. Burch, T. Haldar, S. Sankararaman, B. Pasaniuc, W. J. Gauderman, and J. S. Witte. A two-step approach to testing overall effect of gene–environment interaction for multiple phenotypes. *Bioinformatics*, 36(24): 5640–5648, 2021. doi:10.1093/bioinformatics/btaa1083.

J. D. Malley, J. Kruppa, A. Dasgupta, K. G. Malley, and A. Ziegler. Probability machines: consistent probability estimation using nonparametric learning machines. *Methods of Information in Medicine*, 51(1):74–81, 2012. doi:10.3414/ME00-01-0052.

L. Mentch and G. Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *Journal of Machine Learning Research*, 17 (26):1–41, 2016. URL `http://jmlr.org/papers/v17/14-168.html`.

S. K. Murthy and S. Salzberg. Decision tree induction: How effective is the greedy heuristic? In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, KDD'95, page 222–227. AAAI Press, 1995.

S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994. doi:10.1613/jair.63.

K. K. Nicodemus, J. D. Malley, C. Strobl, and A. Ziegler. The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinformatics*, 11:110, 2010. doi:10.1186/1471-2105-11-110.

V. Ü. Onay, L. Briollais, J. A. Knight, E. Shi, Y. Wang, S. Wells, H. Li, I. Rajendram, I. L. Andrulis, and H. Ozcelik. SNP-SNP interactions in breast cancer susceptibility. *BMC Cancer*, 6:114, 2006. doi:10.1186/1471-2407-6-114.

R. Ottman. Gene–environment interaction: Definitions and study design. *Preventive Medicine*, 25(6):764–770, 1996. doi:10.1006/pmed.1996.0117.

C. J. Patel, J. Bhattacharya, and A. J. Butte. An environment-wide association study (EWAS) on type 2 diabetes mellitus. *PLOS ONE*, 5(5):1–10, 2010. doi:10.1371/journal.pone.0010746.

M. A. Pourhoseingholi, A. R. Baghestani, and M. Vahedi. How to control confounding effects by statistical analysis. *Gastroenterology and Hepatology from Bed to Bench*, 5(2):79–83, 2012. URL `https://journals.sbmu.ac.ir/ghfbb/index.php/ghfbb/article/view/246`.

F. Privé, H. Aschard, and M. G. B. Blum. Efficient implementation of penalized regression for genetic risk prediction. *Genetics*, 212(1):65–74, 2019. doi:10.1534/genetics.119.302019.

P. Probst, M. N. Wright, and A.-L. Boulesteix. Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9 (3):e1301, 2019. doi:10.1002/widm.1301.

F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(3):199–215, 2003. doi:10.1023/A:1024099825458.

S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. De Bakker, M. J. Daly, and S. Pak C. PLINK: A tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007. doi:10.1086/519795.

J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers Inc., San Mateo, California, USA, 1993.

M. D. Ritchie and K. Van Steen. The search for gene-gene interactions in genome-wide association studies: challenges in abundance of methods, practical considerations, and biological interpretation. *Annals of Translational Medicine*, 6(8): 157, 2018. doi:10.21037/atm.2018.04.05.

I. Ruczinski, C. Kooperberg, and M. LeBlanc. Logic regression. *Journal of Computational and Graphical Statistics*, 12(3):475–511, 2003. doi:10.1198/1061860032238.

O. Sangha. Epidemiology of rheumatic diseases. *Rheumatology*, 39(suppl_2):3–12, 2000. doi:10.1093/rheumatology/39.suppl_2.3.

N. Scherer, P. Sekula, P. Pfaffelhuber, and P. Schlosser. pgainsim: an R-package to assess the mode of inheritance for quantitative trait loci in GWAS. *Bioinformatics*, 37(18):3061–3063, 2021. doi:10.1093/bioinformatics/btab150.

T. Schikowski, D. Sugiri, U. Ranft, U. Gehring, J. Heinrich, H.-E. Wichmann, and U. Krämer. Long-term air pollution exposure and living close to busy roads are associated with COPD in women. *Respiratory Research*, 6:152, 2005. doi:10.1186/1465-9921-6-152.

H. Schwender and K. Ickstadt. Identification of SNP interactions using logic regression. *Biostatistics*, 9(1):187–198, 2007. doi:10.1093/biostatistics/kxm024.

M. Segal and Y. Xiao. Multivariate random forests. *WIREs Data Mining and Knowledge Discovery*, 1(1):80–87, 2011. doi:10.1002/widm.12.

L. S. Shapley. A value for n-person games. In *Contributions to the Theory of Games, Volume II*, pages 307–317. Princeton University Press, Princeton, NJ, USA, 1953. doi:10.1515/9781400881970-018.

S. Shi, N. Yuan, M. Yang, Z. Du, J. Wang, X. Sheng, J. Wu, and J. Xiao. Comprehensive assessment of genotype imputation performance. *Human Heredity*, 83(3):107–116, 2019. doi:10.1159/000489758.

M. Slatkin. Linkage disequilibrium – understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics*, 9(6):477–485, 2008. doi:10.1038/nrg2361.

D. Sorokina, R. Caruana, M. Riedewald, and D. Fink. Detecting statistical interactions with additive groves of trees. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1000–1007, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1390156.1390282.

S. Stringer, N. R. Wray, R. S. Kahn, and E. M. Derks. Underestimated effect sizes in GWAS: Fundamental limitations of single SNP analysis for dichotomous phenotypes. *PLOS ONE*, 6(11):1–7, 2011. doi:10.1371/journal.pone.0027964.

Y.-R. Su, C.-Z. Di, L. Hsu, Genetics, and E. of Colorectal Cancer Consortium. A unified powerful set-based test for sequencing data analysis of GxE interactions. *Biostatistics*, 18(1):119–131, 2016. doi:10.1093/biostatistics/kxw034.

C. Sudlow, J. Gallacher, N. Allen, V. Beral, P. Burton, J. Danesh, P. Downey, P. Elliott, J. Green, M. Landray, B. Liu, P. Matthews, G. Ong, J. Pell, A. Silman, A. Young, T. Sprosen, T. Peakman, and R. Collins. UK Biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLOS Medicine*, 12(3):1–10, 2015. doi:10.1371/journal.pmed.1001779.

J. H. Sul, L. S. Martin, and E. Eskin. Population structure in genetic studies: Confounding factors and mixed models. *PLOS Genetics*, 14(12):1–22, 2018. doi:10.1371/journal.pgen.1007309.

N. Tesi, S. J. van der Lee, M. Hulsman, I. E. Jansen, N. Stringa, N. M. van Schoor, P. Scheltens, W. M. van der Flier, M. Huisman, M. J. T. Reinders, and H. Holstege. Polygenic risk score of longevity predicts longer survival across an age continuum. *The Journals of Gerontology: Series A*, 76(5):750–759, 2020. doi:10.1093/gerona/glaa289.

The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015. doi:10.1038/nature15393.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. doi:10.1111/j.2517-6161.1996.tb02080.x.

T. Tietz, S. Selinski, K. Golka, J. G. Hengstler, S. Gripp, K. Ickstadt, I. Ruczinski, and H. Schwender. Identification of interactions of binary variables associated with survival time using survivalFS. *Archives of Toxicology*, 93(3):585–602, 2019. doi:10.1007/s00204-019-02398-6.

P. R. Timmers, N. Mounier, K. Lall, K. Fischer, Z. Ning, X. Feng, A. D. Bretherick, D. W. Clark, eQTLGen Consortium, X. Shen, T. Esko, Z. Kutalik, J. F. Wilson, and P. K. Joshi. Genomics of 1 million parent lifespans implicates novel pathways and common diseases and distinguishes survival chances. *eLife*, 8:e39856, 2019. doi:10.7554/eLife.39856.

A. Torkamani, N. E. Wineinger, and E. J. Topol. The personal and clinical utility of polygenic risk scores. *Nature Reviews Genetics*, 19(9):581–590, 2018. doi:10.1038/s41576-018-0018-x.

E. Uffelmann, Q. Q. Huang, N. S. Munung, J. De Vries, Y. Okada, A. R. Martin, H. C. Martin, T. Lappalainen, and D. Posthuma. Genome-wide association studies. *Nature Reviews Methods Primers*, 1:59, 2021. doi:10.1038/s43586-021-00056-9.

J. Vanhoof, K. Declerck, and P. Geusens. Prevalence of rheumatic diseases in a rheumatological outpatient practice. *Annals of the Rheumatic Diseases*, 61(5): 453–455, 2002. doi:10.1136/ard.61.5.453.

D. S. Watson and M. N. Wright. Testing conditional independence in supervised learning algorithms. *Machine Learning*, 110:2107–2129, 2021. doi:10.1007/s10994-021-06030-6.

C. P. Wild. Complementing the genome with an “exposome”: The outstanding challenge of environmental exposure measurement in molecular epidemiology. *Cancer Epidemiology, Biomarkers & Prevention*, 14(8):1847–1850, 2005. doi:10.1158/1055-9965.EPI-05-0456.

M. Woodward. *Epidemiology: Study Design and Data Analysis*. Chapman and Hall/CRC, Boca Raton, FL, USA, 2013. doi:10.1201/b16343.

N. R. Wray, T. Lin, J. Austin, J. J. McGrath, I. B. Hickie, G. K. Murray, and P. M. Visscher. From basic science to clinical application of polygenic risk scores: A primer. *JAMA Psychiatry*, 78(1):101–109, 2021. doi:10.1001/jamapsychiatry.2020.3049.

M. N. Wright, A. Ziegler, and I. R. König. Do little interactions get lost in dark random forests? *BMC Bioinformatics*, 17:145, 2016. doi:10.1186/s12859-016-0995-8.

Y. Xiao, M. A. Taub, I. Ruczinski, F. Begum, J. B. Hetmanski, H. Schwender, E. J. Leslie, D. C. Koboldt, J. C. Murray, M. L. Marazita, and T. H. Beaty. Evidence for SNP-SNP interaction identified through targeted sequencing of cleft case-parent trios. *Genetic Epidemiology*, 41(3):244–250, 2017. doi:10.1002/gepi.22023.

A. I. Yashin, I. Zhbannikov, L. Arbeeva, K. G. Arbeev, D. Wu, I. Akushevich, A. Yashkin, M. Kovtun, A. M. Kulminski, E. Stallard, I. Kulminskaya, and S. Ukraintseva. Pure and confounded effects of causal SNPs on longevity: Insights for proper interpretation of research findings in GWAS of populations with different genetic structures. *Frontiers in Genetics*, 7, 2016. doi:10.3389/fgene.2016.00188.

W. Yoo, B. A. Ference, M. L. Cote, and A. Schwartz. A comparison of logistic regression, logic regression, classification tree, and random forests to identify effective gene-gene and gene-environmental interactions. *International Journal of Applied Science and Technology*, 2(7):268, 2012.

G. Yu, J. Bien, and R. Tibshirani. Reluctant interaction modeling. *arXiv*, 2019. doi:10.48550/ARXIV.1907.08414.

E. Zanelli, F. C. Breedveld, and R. R. P. de Vries. Hla class II association with rheumatoid arthritis: Facts and interpretations. *Human Immunology*, 61(12): 1254–1261, 2000. doi:10.1016/S0198-8859(00)00185-3.

A. Zeileis, T. Hothorn, and K. Hornik. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514, 2008. doi:10.1198/106186008X319331.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67 (2):301–320, 2005. doi:10.1111/j.1467-9868.2005.00503.x.

Eidesstattliche Versicherung

Ich versichere an Eides statt, dass die Dissertation von mir selbstständig und ohne unzulässige fremde Hilfe unter Beachtung der Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf erstellt worden ist.

Michael Lau, Februar 2024, Düsseldorf