frontiers | Frontiers in Genetics

TYPE Methods
PUBLISHED 10 January 2023
DOI 10.3389/fgene.2022.1076440

OPEN ACCESS

EDITED BY
Angelo Facchiano,
National Research Council (CNR), Italy

REVIEWED BY
Wenan Chen,
St. Jude Children's Research Hospital, United States
Tianyuan Lu,
McGill University, Canada

*CORRESPONDENCE
Hannah Klinkhammer,
klinkhammer@imbie.uni-bonn.de

SPECIALTY SECTION
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

RECEIVED 21 October 2022
ACCEPTED 20 December 2022
PUBLISHED 10 January 2023

CITATION
Klinkhammer H, Staerk C, Maj C, Krawitz PM and Mayr A (2023), A statistical boosting framework for polygenic risk scores based on large-scale genotype data.
Front. Genet. 13:1076440.
doi: 10.3389/fgene.2022.1076440

A statistical boosting framework for polygenic risk scores based on large-scale genotype data

Hannah Klinkhammer[1,2]*, Christian Staerk[1], Carlo Maj[2,3], Peter Michael Krawitz[2] and Andreas Mayr[1]

[1]Institute for Medical Biometry, Informatics and Epidemiology, Medical Faculty, University of Bonn, Bonn, Germany, [2]Institute for Genomic Statistics and Bioinformatics, Medical Faculty, University of Bonn, Bonn, Germany, [3]Center for Human Genetics, University of Marburg, Marburg, Germany

Polygenic risk scores (PRS) evaluate the individual genetic liability to a certain trait and are expected to play an increasingly important role in clinical risk stratification. Most often, PRS are estimated based on summary statistics of univariate effects derived from genome-wide association studies. To improve the predictive performance of PRS, it is desirable to fit multivariable models directly on the genetic data. Due to the large and high-dimensional data, a direct application of existing methods is often not feasible and new efficient algorithms are required to overcome the computational burden regarding efficiency and memory demands. We develop an adapted component-wise L_2-boosting algorithm to fit genotype data from large cohort studies to continuous outcomes using linear base-learners for the genetic variants. Similar to the snpnet approach implementing lasso regression, the proposed snpboost approach iteratively works on smaller batches of variants. By restricting the set of possible base-learners in each boosting step to variants most correlated with the residuals from previous iterations, the computational efficiency can be substantially increased without losing prediction accuracy. Furthermore, for large-scale data based on various traits from the UK Biobank we show that our method yields competitive prediction accuracy and computational efficiency compared to the snpnet approach and further commonly used methods. Due to the modular structure of boosting, our framework can be further extended to construct PRS for different outcome data and effect types—we illustrate this for the prediction of binary traits.

KEYWORDS

polygenic risk score (PRS), high-dimensional data, variable selection, boosting, GWAS—genome-wide association study, prediction

1 Introduction

In times of next-generation sequencing and decreasing costs for whole genome sequencing, the amount of available genotype data has increased dramatically in recent years, giving rise to new genetic insights (Beesley et al., 2020; National Human Genome Research Institute, 2021).

Polygenic risk scores (PRS) measure the individual genetic liability to a certain trait and can provide relevant information in the context of disease-risk stratification. In contrast to high-impact monogenic variants, which are mostly rare and have a high effect size, PRS are derived from common variants such as single-nucleotide polymorphisms (SNPs) with low or medium effect sizes. Polygenic effects could also explain part of the incomplete penetrance seen in many identified monogenic variants, as for example in the genes BRCA1 and BRCA2 both leading to a highly increased risk of breast cancer (Kuchenbaecker et al., 2017). Recent studies on the UK Biobank suggest that high-impact monogenic variants, PRS and family history could contribute

additively to the risk of developing breast and prostate cancer (Hassanin et al., 2021). Despite these findings, PRS still lack to explain relevant parts of the estimated heritability of many traits.

PRS are typically derived as a sum of risk allele counts weighted by univariate effect estimates of the measured variants based on summary statistics from genome-wide association studies (GWAS) (Choi et al., 2020). Despite several approaches to account for linkage disequilibrium (LD, referring to the correlation structure between variants) and for the selection of informative variants (Euesden et al., 2014; Vilhjálmsson et al., 2015; Mak et al., 2017; Privé et al., 2021), the univariate structure of the estimation cannot fully account for interdependencies between the variants. For example, lassosum (Mak et al., 2017) adopts an L_1 penalty term and solves a lasso-like problem while only using summary statistics and a LD reference panel. However, as published summary statistics and LD reference panels are most often based on different samples, lassosum can generally only approximate the full lasso path. A natural extension of using effect estimates from univariate models could hence be to fit a single multivariable model. While this approach seems natural from a methodological perspective, a direct application of existing methods is typically infeasible due to the high dimensionality of the genotype data, which can easily exceed the available computer memory. Recently, some approaches have been proposed to overcome this computational burden (Privé et al., 2018; Qian et al., 2020; Maj et al., 2022). In particular, Qian et al. proposed the so-called batch screening iterative lasso (BASIL) algorithm to fit the lasso on the complete original genotype data (Tibshirani, 1996; Qian et al., 2020; Li et al., 2022). The algorithm works on subsets of variants and computes the complete lasso path in an iterative fashion. Apart from the lasso, the algorithm can also be extended to other penalized regression methods such as the relaxed lasso (Meinshausen, 2007) or the elastic net (Zou and Hastie, 2005). In this context, Qian et al. were able to demonstrate that multivariable regularized PRS models fitted *via* the BASIL algorithm outperform the classical GWAS-based PRS for various traits such as height and high cholesterol.

While penalized regression models like the lasso and the elastic net impose explicit regularization, statistical boosting represents an alternative approach by introducing an implicit algorithmic regularization when combined with early stopping (Bühlmann and Hothorn, 2007; Mayr and Hofner, 2018). Boosting algorithms iteratively fit pre-defined base-learners to the gradient of the loss function, selecting the most influential base-learner in each step. The main tuning parameter of boosting algorithms is the number of iterations, which enables implicit variable selection and leads to sparse models. Due to its modular structure, boosting allows to combine possible base-learners with any convex loss function. These algorithms hence offer a great flexibility for statistical modelling, including various response types and the estimation of non-linear or other types of effects. A recent work has incorporated boosting into PRS modelling *via* a three-step approach (Maj et al., 2022): First, a marginal screening approach was applied on all variants to identify potentially informative ones. Then, multivariable algorithms including probing with boosting (Thomas et al., 2017) were applied on blocks of variants in LD to select ("fine-map") the most informative variants. Finally, a statistical boosting model was fitted on the variant set created by joining the selected variants of all chunks. This approach yielded particularly sparse and interpretable models, whose predictive performance was superior to PRS derived by univariate methods like clumping and thresholding (Euesden et al., 2014) and was outperformed by the predictive performance achieved by the lasso *via* the BASIL algorithm. However this approach includes pre-filtering of the variants and is computationally demanding.

In this article we introduce a new framework to boost PRS, starting with a new computational approach to build L_2-boosting models on large-scale genotype data for quantitative traits. Similar to the snpnet approach for the lasso, our algorithm iteratively works on smaller batches of variants. Yet, in contrast to recent boosting methods (Staerk and Mayr, 2021; Maj et al., 2022), the variants do not need to be pre-filtered in our snpboost approach and the batches are not pre-defined or randomly sampled, but chosen iteratively and deterministically in a data-driven way based on the correlations of the variants to the remaining residuals. By restricting the set of available base-learners in each step to those variants which were most correlated with residuals from a previous iteration, we are able to reduce the search space and decrease the computational time compared to a classical component-wise boosting algorithm.

We conducted a simulation study to examine the performance of our adapted boosting algorithm snpboost compared to the original L_2-boosting on a reduced but still high-dimensional data set, on which the application of standard L_2-boosting was still computationally feasible. Furthermore, we simulated data of higher dimensionality and larger sample size to investigate the influence of various hyperparameters (including the batch size) on the prediction accuracy and computational burden of the snpboost approach in a typical large-scale setting. We discuss reasonable default values for the hyperparameters which are incorporated in the provided R implementation (https://github.com/hklinkhammer/snpboost). Finally, we constructed multivariable PRS for various traits on data from the UK Biobank *via* application of snpboost and compared the performance of our approach to the lasso estimates from the BASIL algorithm proposed by Qian et al. as well as to further commonly used methods. On the examined phenotypes we found highly comparable predictive performance while our adapted boosting approach had a tendency to select sparser models compared to the lasso and the other methods. Finally, we illustrate how the framework can be conveniently extended to the classification of binary phenotypes by the incorporation of different loss functions.

2 Methods

For $n \in \mathbb{N}$ individuals, let $\mathbf{y} = (y_1, \ldots, y_n)' \in \mathbb{R}^n$ denote a particular continuous phenotype of interest. Furthermore, let X_j correspond to the genetic variant j, for $j = 1, \ldots, p$. The observed dosage data of n individuals is given in the genotype matrix $\mathbf{X} = (x_{i,j}) \in [0, 2]^{n \times p}$, where $\mathbf{x}_j \in [0, 2]^n$ corresponds to the jth column of $\mathbf{X}$. We consider a linear regression model

$$\mathbb{E}(y_i|\mathbf{X}) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}, \quad i = 1, \ldots, n, \tag{1}$$

With coefficients $\beta_0 \in \mathbb{R}$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)' \in \mathbb{R}^p$. The aim is to determine coefficients $\hat{\beta}_0 \in \mathbb{R}$ and $\hat{\boldsymbol{\beta}} \in \mathbb{R}^p$ such that the estimator $\hat{\mathbf{y}} = \hat{\beta}_0 + \mathbf{X}\hat{\boldsymbol{\beta}}$ minimizes the mean squared error of prediction on an independent test set MSEP $= \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (\hat{y}_{\text{test},i} - y_{\text{test},i})^2$. Additionally, one is often interested in relatively sparse models in the sense that only a fraction of the coefficient vector $\hat{\boldsymbol{\beta}} \in \mathbb{R}^p$ is non-zero.

In high-dimensional settings with $p > n$ it is not feasible to apply classical estimation techniques like the ordinary least squares estimator. A commonly-used solution is to consider further constraints on the coefficient vector resulting in penalized regression methods including the lasso (Tibshirani, 1996). The lasso incorporates an L_1-penalty on the coefficient vector such that the lasso estimate $\hat{\boldsymbol{\beta}}^{\text{lasso}}$ is given by

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \text{argmin}_{(\beta_0,\boldsymbol{\beta})}\left[\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right] \quad (2)$$

for some $\lambda \geq 0$. The explicit L_1-penalization of the coefficient vector leads to shrinkage of the coefficient estimates. In contrast to ridge regression (Hoerl and Kennard, 2000), the use of the L_1-penalty enables to set some parameters exactly to zero corresponding to sparse models. There has been extensive research on the theoretical properties of the lasso including oracle inequalities in high-dimensional settings (e.g., Fu and Knight (2000); Greenshtein and Ritov (2004); Bunea et al. (2007); van de Geer (2008)). Nevertheless, there are situations leading to variable selection problems of the lasso, particularly in the presence of high correlations between signal and noise variables (Hepp et al., 2016). When working with genotype data, high correlations between signal and noise variables might often be present as a result of LD, i.e., genetic variants that have close positions on the DNA strand tend to be highly correlated.

An alternative to explicitly penalized regression methods such as the lasso is statistical gradient boosting (Bühlmann and Hothorn, 2007; Mayr and Hofner, 2018). Gradient boosting requires the specification of a loss function $f(\boldsymbol{y}, \hat{\boldsymbol{y}})$ and so-called base-learners h_j that are iteratively fitted to the response. In detail, the aim is again to fit the linear regression model (1) which is performed in an iterative fashion. Starting at iteration $m = 0$ with a starting value $\hat{\boldsymbol{y}}^{(0)} = \mathbf{0}$, the following steps are repeated until a maximum number m_{stop} of boosting iterations is reached (Bühlmann and Hothorn, 2007):

1. Set $m := m+1$ and compute the negative gradient vector of the loss function:

$$\boldsymbol{u}^{(m)} = -\frac{\partial f(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \hat{\boldsymbol{y}}}\bigg|_{\hat{\boldsymbol{y}}=\hat{\boldsymbol{y}}^{(m-1)}}$$

2. Fit every base-learner h_j separately to the negative gradient vector $\boldsymbol{u}^{(m)}$ and select the best fitting base-learner $\hat{h}_{j^*}^{(m)}(X_j)$.
3. Update the predictor with the learning rate $0 \leq \nu \leq 1$: $\hat{\boldsymbol{y}}^{(m)} = \hat{\boldsymbol{y}}^{(m-1)} + \nu\hat{h}_{j^*}^{(m)}(X_j)$
4. Stop if $m = m_{\text{stop}}$.

Stopping the algorithm before it converges (early stopping) leads to implicit regularization and shrinkage of effect estimates. The component-wise L_2-boosting algorithm (Bühlmann and Yu, 2003; Bühlmann and Hothorn, 2007) employs the squared error $f(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2$ as a loss function (Bühlmann and Yu, 2003) and separate univariate linear regression models of the residuals on the jth genetic variant as base-learners (i.e., $h_j(X_j) = \beta_0 + \beta_j X_j$, for $j = 1, \ldots, p$). In low-dimensional ($p < n$) settings this set-up mimics a classical Gaussian linear model and converges to the least squares solution for large values of m_{stop}. The general boosting procedure can be interpreted as gradient descent in function space, where the residual vector represents the gradient of the L_2 loss and the function space is provided by the different base-learner solutions (Friedman, 2001; Bühlmann and Yu, 2003; Mayr and Hofner, 2018). The previously described steps transform therefore into the following procedure (shown in grey in Figure 1): The best fitting base-learner in boosting step m+1 corresponds to the variant j^* with the highest Pearson correlation $\rho(\boldsymbol{x}_{j^*}, \boldsymbol{r}^{(m)})$ to the residuals $\boldsymbol{r}^{(m)} = \boldsymbol{y} - \hat{\boldsymbol{y}}^{(m)}$ resulting from the previous boosting step m. We then fit a linear regression model of the current residuals $\boldsymbol{r}^{(m)}$ on the variant j^* and update the corresponding coefficient $\hat{\beta}_{j^*}^{(m+1)}$ as well as the intercept $\hat{\beta}_0^{(m+1)}$. This is repeated until a maximum number of boosting iterations is reached or any other early stopping criterion is fulfilled. If additional covariates apart from the genetic variants are included in the model, they are treated as mandatory covariates—similar to the intercept. The additional covariates are included in each single base-learner and are hence updated in each boosting step without competing with the genetic variants.

Hepp et al. (2016) investigated the commonalities and differences between the lasso and statistical boosting: while there are (low-dimensional) settings in which the gradient boosting approximates the lasso coefficient paths arbitrarily close when the learning rate ν is approaching 0, their results generally differ if the coefficient paths are not monotone. The authors note that, in contrast to the lasso which limits the sum of the absolute values of the coefficients for each penalty parameter λ separately, boosting limits the total L_1-arc-length of all coefficient curves (Hepp et al., 2016). Interpreting this as the total absolute distance "travelled" by all coefficients among the coefficient paths through the iterations $m = 1, \ldots, m_{\text{stop}}$, it becomes clear that the solution in a certain iteration depends on all previous solutions of the iterative algorithm. This might lead to more stable pathways particularly in settings with high correlations between independent variables, which is typical for genetic data. Hepp et al. conducted several numerical experiments including high-dimensional settings in which they found similar predictive performance of lasso and boosting. In detail, boosting tended to yield slightly better prediction results while the lasso tended to result in sparser models with faster computations. On the other hand, the boosting algorithm can be easily extended to different response types as well as to different effects, including non-linear and interaction effects. In terms of genetic data, interaction effects can be used to model and identify epistatic effects and gene-environment interactions.

When working on genetic data from large cohort studies we do not only face a high-dimensional setting with $p > n$ but also a large-scale setting with large sample sizes n *and* large numbers of variants p. Large-scale settings often lead to extended computational times as well as memory issues. To overcome these and apply statistical boosting on genotype data, we implemented an adapted component-wise L_2-boosting algorithm that is built on the snpnet framework (Qian et al., 2020) and works on batches of variants. To do so, we additionally incorporate a batch-building step before starting the boosting iterations (shown in blue in Figure 1). In this step we extract the p_{batch} variants ($p_{\text{batch}} \ll p$) with the highest correlation $\rho(\boldsymbol{x}_j, \boldsymbol{r}^{(m)})$ to the current residual vector and include them in the batch B_k. A maximum number of m_{batch} boosting iterations is performed on batch B_k before the next batch is built based on the correlations of all p variants to the updated residuals. In total, we fit a maximum of $b_{\max}$ batches or stop early if an early stopping criterion is fulfilled. The algorithm is summarised in Table 1 and Figure 1.

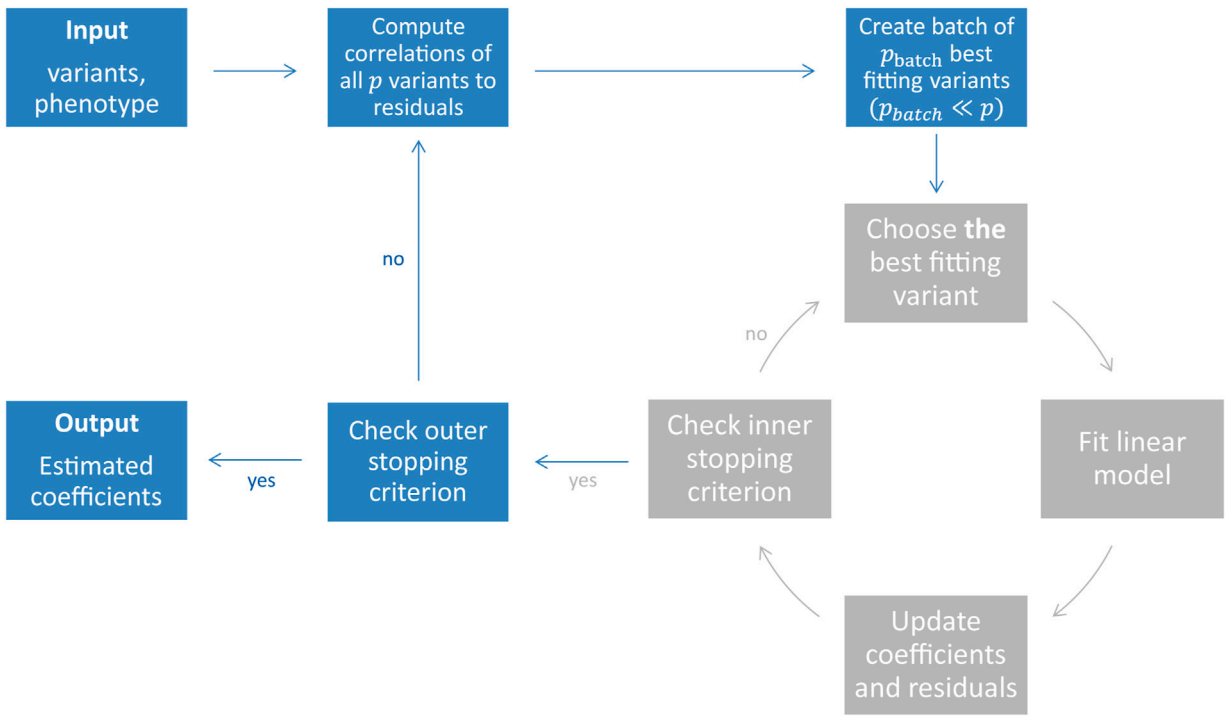

FIGURE 1
Illustration of the snpboost algorithm. The snpboost algorithm consists of an outer loop to create batches (shown in blue) and an inner loop representing the boosting on one batch (shown in grey).

TABLE 1 Definition of the snpboost algorithm without additional covariates. If additional covariates apart from the genetic variants should be included in the model, they are treated as mandatory covariates—similar to the intercept. The additional covariates are included in each single base-learner and are hence updated in each boosting step without competing with the genetic variants.

Algorithm: snpboost

Input: phenotype data $\mathbf{y} \in \mathbb{R}^n$, genotype data $\mathbf{X} \in [0,2]^{n \times p}$,
batch size $p_{\text{batch}} \in \{1, \ldots, p\}$,
learning rate $\nu > 0$,
maximum number of boosting iterations per batch $m_{\text{batch}} \in \mathbb{N}$,
maximum number of batches $b_{\max} \in \mathbb{N}$,
stopping lag for outer stopping criterion $b_{\text{stop}} \in \mathbb{N}$.

Algorithm:

1. **Initialization**: Set boosting index $m = 0$,
residuals $\mathbf{r}^{(0)} = \mathbf{y} - \bar{\mathbf{y}}$,
coefficients $\hat{\beta}_0^{(0)} = \bar{\mathbf{y}}$, $\hat{\beta}_j^{(0)} = 0$, $j = 1, \ldots, p$.
2. **Outer loop**: Set outer counter $k = 1$.
   a. Screening: Compute correlations $c_j^{(m)} = \rho(\mathbf{r}^{(m)}, \mathbf{x}_j)$, $j = 1, \ldots, p$.
   Create batch B_k of p_{batch} variants with highest absolute correlations $|c_j^{(m)}|$.
   Save the highest absolute correlation outside the batch $c_{\text{stop}} = \max_{j \notin B_k} |c_j^{(m)}|$.
   b. **Inner loop**: Set inner counter $l = 1$.
   (1) If $l > 1$, compute correlations inside batch: $c_j^{(m)} = \rho(\mathbf{r}^{(m)}, \mathbf{x}_j), \quad j \in B_k$
   (2) Choose variant j^* with the highest absolute correlation $|c_{j^*}^{(m)}| = \max_{j \in B_k} |c_j^{(m)}|$.
   If the current maximum absolute correlation inside the batch is smaller than the highest correlation outside the batch, i.e. if $|c_{j^*}^{(m)}| < c_{\text{stop}}$, stop the inner loop; else set $m := m + 1$.
   (3) Fit linear model: $\mathbb{E}(r^{(m-1)} \mid \mathbf{X}) = \hat{\beta}_0 + \hat{\beta}_{j^*} X_{j^*}$
   (4) Update coefficients with learning rate ν and $\hat{\boldsymbol{\beta}}$ from iii.:
   $\hat{\beta}_0^{(m)} = \hat{\beta}_0^{(m-1)} + \nu \hat{\beta}_0, \quad \hat{\beta}_{j^*}^{(m)} = \hat{\beta}_{j^*}^{(m-1)} + \nu \hat{\beta}_{j^*},$
   $\hat{\beta}_j^{(m)} = \hat{\beta}_j^{(m-1)}, \quad j \in \{1, \ldots, p\} \backslash \{j^*\}$
   as well as residuals $\mathbf{r}^{(m)} = \mathbf{y} - \hat{\beta}_0^{(m)} - \mathbf{X}\hat{\boldsymbol{\beta}}^{(m)}$
   (5) If $l = m_{\text{batch}}$, end inner loop, else increase inner counter $l := l + 1$.
   c. If $k = b_{\max}$ or if the MSEP on the validation set has not decreased for b_{stop} batches, stop the outer loop; else increase the outer counter $k := k + 1$.
3. **Final model choice:** Find $m_{\text{stop}} \in \{1, \ldots, m\}$ corresponding to the lowest MSEP on the validation set. The final coefficient estimates are given by $\hat{\beta}_0^{(m_{\text{stop}})} \in \mathbb{R}$ and $\hat{\boldsymbol{\beta}}^{(m_{\text{stop}})} \in \mathbb{R}^p$.

By iteratively working on batches of variants we save computational time and memory because only parts of the variants have to be loaded into memory at once. Additionally, not every step requires the calculation of all potential base-learner solutions and the updated correlations for all variants. By this, we encourage additional sparsity by restricting the search space in terms of the set of available base-learners (as variants not included in the current batch cannot be selected). To examine when a new set of base-learners should be considered, which corresponds to the question when to stop the inner loop (inside the batches) and create a new batch of variants, we incorporated another step: we monitor the correlations of the variants inside a batch to the residuals and compare them to the correlations of variants outside of the batch. When creating a batch B_k we therefore compute and store the highest outer correlation $c_{\text{stop}} := \max_{j \notin B_k} |\rho(\mathbf{r}^{(m)}, \mathbf{x}_j)|$. After each boosting step m we check if the greatest absolute correlation of the variants inside the batch B_k to the current residual vector $\mathbf{r}^{(m)}$ is smaller than c_{stop}:

$$c_{\text{stop}} > \max_{j \in B_k} \left| \rho\left(\mathbf{r}^{(m)}, \mathbf{x}_j\right) \right|. \tag{3}$$

If inequality Eq. 3 holds true, we stop the inner loop and create a new batch since a variant outside the batch may provide a better fit to the current residual vector. In the original L_2-boosting without batches, the variant with the highest correlation to the residuals would be chosen in each boosting step. The incorporation of batches in general limits this choice to the variants inside the batch. However, the proposed stopping criterion provides an indication to consider variants outside the batch which may be higher correlated with the current residuals. Actually, if all variants were independent, the proposed stopping criterion would lead to the same choice of variants in each boosting step in snpboost as in the original L_2-boosting. Despite LD, our simulation results show that the

proposed stopping criterion yields reasonable variant choices and results in a competitive predictive performance (Section 3.1.1). Additionally, the inner loop is also stopped if the number of updates inside the batch reaches m_{batch}.

Furthermore, we need to determine after how many batches the algorithm should terminate. In classical statistical boosting the number of boosting iterations is often selected by cross-validation or resampling techniques—mimicking an additional data set to validate the predictive performance of the resulting models. However, if the data set is large enough, one can also directly divide the data into training and validation set. As in Qian et al. (2020), we hence simultaneously monitor the predictive performance of our model on an independent validation set while fitting on the training set. As a validation criterion for the predictive performance we use the MSEP on the validation set. The outer loop consisting of the batch-building step is stopped if the MSEP on the validation set has not decreased for b_{stop} batches or after a maximum number of b_{max} batches have been processed.

The proposed method is implemented as an add-on to the snpnet package by Qian et al. (2020) in the statistical computing environment R (R Core Team (2021), https://github.com/hklinkhammer/snpboost). While we are also incorporating PLINK 2.0 (Chang et al., 2015) to compute the correlations and build the batches in the outer loop, we replaced the fitting of the lasso by the adapted component-wise L_2-boosting algorithm on the resulting batches (Table 1; Figure 1).

3 Empirical results

3.1 Simulation study

We conducted a simulation study to investigate the behaviour of the proposed snpboost algorithm in various controlled data scenarios. The simulation study aims at two main goals: first, to examine potential differences in performance compared to the original component-wise L_2-boosting (Bühlmann and Yu, 2003) in smaller settings and, second, to gain insights on how to choose the included hyperparameters in practical situations.

Simulations are based on the UK Biobank genotype data (Bycroft et al., 2018) obtained under application number 81202 combined with simulated phenotypes. We restricted the individuals to white British ancestry and used the PLINK 2.0 function–thin-indiv-count to randomly sample n individuals, of which 50%, 20% and 30% were assigned to the training, validation and test set, respectively (Chang et al., 2015; Purcell and Chang, 2015). Then, p variants with minor allele frequency not less than 1% were randomly sampled using PLINK 2.0's–thin-count. Missing genotypes were replaced by the reference allele using the R package bigsnpr (Privé et al., 2018).

Continuous phenotypes were simulated from a linear model with Gaussian distributed noise and effect sizes using bigsnpr. To account for different genetic architectures, we considered varying heritability h^2 and sparsity s, defined as the amount of variance explained by the genetic liability and the proportion of causal variants, respectively. For each setting of h^2 and s, we simulated 100 different datasets. PRS models were derived by snpboost and evaluated by using various metrics regarding the predictive performance and the accuracy of the estimated coefficients. In detail, the predictive performance was measured by the MSEP and the R^2 value defined as the squared correlation between the predicted and the true phenotype on the independent test set. To assess the computational efficiency we measured the computation time of the algorithm. The accuracy of the resulting estimates was evaluated by the number of included variants in the final model and the mean squared error (MSE) of the estimates as well as the true positive (TP) rates and precision regarding variant selection. Additional results for all considered settings as well as comparisons to snpnet can be found in the Supplementary Material (Supplementary Figures S1–S6).

3.1.1 Comparison to original L_2-boosting in smaller settings

To analyse the performance of snpboost compared to the original component-wise L_2-boosting algorithm (Bühlmann and Yu, 2003), we used a single large batch with batch size $p_{batch} = p$ in the snpboost algorithm on simulated data with reduced dimensionality. We then compared the results to the ones derived by using smaller batches in terms of predictive performance, computation time, mean squared errors of the estimated coefficients as well as true positive rates and precision regarding variant selection. The simulations were conducted for n = 20,000 observations (10,000 training set, 4,000 validation set, 6,000 test set) and p = 20,000 variants as well as for varying degrees of heritability and sparsity. To obtain comparable results we chose a fixed number of boosting iterations independent of the batch size p_{batch} and a fixed learning rate $\nu = 0.1$. For each simulation, 10 CPUs with 1 GB memory each were used.

Figure 2 displays the boxplots of each metric obtained after 1,500 boosting iterations for heritability h^2 = 50% and sparsity s = 0.1%, i.e., 20 influential variants. Incorporating batches did not largely affect the predictive performance in terms of R^2 and MSEP nor the MSE of the coefficient estimates (MSE results not shown). However, different batch sizes do not always yield the same models as L_2-boosting as can be observed from the number of variants included in the final models. The models resulting from a batch size of p_{batch} = 1,000 tend to contain less variants than the ones from the original L_2-boosting (batch size p_{batch} = 20,000). This could be explained by the reduced search space in each boosting step and a trade-off between exploration (genome-wide search) and exploitation (search inside the batch). As a consequence, variants within the batch that are already in the model are more often updated instead of including new variants outside of the batch. The same holds true when comparing the number of chosen variants for batch size p_{batch} = 1,000 to smaller batch sizes (i.e., p_{batch} = 10 and p_{batch} = 100). As all models tend to overestimate the number of influential variants, the lower number of selected variants for batch size p_{batch} = 1,000 corresponds to a higher precision since less false positives are included. The fact that the other metrics remain almost constant suggests that either only variants with very small effects are not included when using a larger batch size or the variants that are updated are highly correlated with the ones not included. Furthermore, incorporating batches in the algorithm has a major effect on the computation time. To interpret the results shown in Figure 2 it is important to understand the two drivers of the computation time. On the one hand, it increases with the number of correlations that have to be calculated in each boosting step which explains the increased computation time of the original L_2-boosting (i.e., a batch size of p_{batch} = 20,000 and 20,000 computed correlations in each boosting step) compared to smaller batch sizes such as p_{batch} = 100 and p_{batch} = 1,000. On the other hand, reading the genotype data from disk when building the batches also increases the

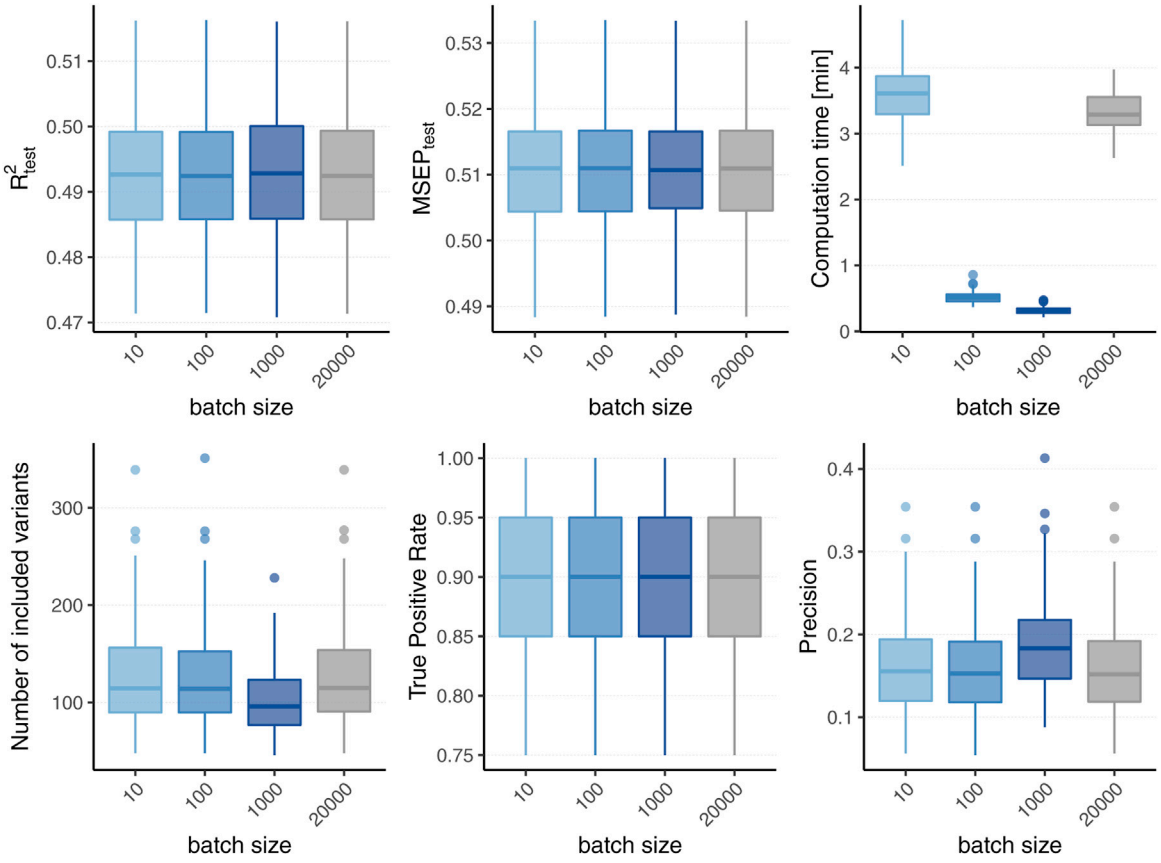

FIGURE 2
Comparison to original L_2-boosting. Results of 100 simulated phenotypes with heritability h^2 = 50% and sparsity s = 0.1% for p = 20,000 variants and n = 20,000 individuals (divided into 50% training, 20% validation and 30% test set). Boxplots of the evaluation metrics obtained after 1,500 boosting iterations are shown depending on the batch size. Batch size p_{batch} = 20,000 corresponds to the original L_2-boosting (shown in grey).

computation time leading to a higher computation time for smaller batches with p_{batch} = 10 for which more reads-from-disk have to be carried out. The varying computation times therefore reflect a trade-off between the number of correlations computed in each boosting step and the number of created batches.

In summary, the incorporation of batches in the boosting algorithm did not affect the predictive performance of the model in our scenarios, while computation time was substantially reduced. However, snpboost does not always lead to the same models as the original L_2-boosting algorithm, in particular in terms of the included variants and sparsity. The results for further settings with different heritability and sparsity were comparable and can be found in the Supplementary Material.

3.1.2 Choice of hyperparameters for large-scale applications

The proposed snpboost algorithm includes various hyperparameters, namely the batch size p_{batch}, the learning rate ν, the maximum number of boosting iterations per batch m_{batch}, the maximum number of processed batches b_{max} and the stopping lag for the outer early stopping criterion b_{stop}. In this section we discuss default values for the hyperparameters to facilitate the applicability of the algorithm in practice. The majority of these parameters do not need to be tuned but can be specified with reasonable default values, e.g., based on results from the literature and experience with the original boosting algorithm. For the remaining ones (p_{batch} and b_{stop}) we examine how they influence the computational and predictive performance of snpboost in a simulation study.

The choice of the learning rate ν can be leaned on widely-used boosting algorithms. A rather small learning rate prevents boosting algorithms from overfitting on single base-learners and is therefore favorable regarding predictive performance. Nevertheless, a smaller learning rate will increase the number of needed boosting iterations to fit the full effect of the base-learners and simultaneously increase the algorithm's computation time. Widely used R packages such as mboost (Bühlmann and Hothorn, 2007; Hothorn et al., 2010) and xgboost (Chen and Guestrin, 2016) use default learning rates of 0.1 and 0.3, respectively. As the effect of the learning rate will be comparable in the proposed adapted boosting algorithm, we decided to specify a fixed default value of $\nu = 0.1$ in all our simulations. For the batch-related hyperparameters we varied the batch size p_{batch} over a range of possible values namely $p_{batch} \in \{10, 100, 1{,}000, 5{,}000\}$ to analyse its effect. For each batch we allow a maximum number of boosting iterations m_{batch} equivalent to the batch size p_{batch}. Since we specified the learning rate with a rather small fixed value and due to the correlation-based early stopping criterion, this choice should prevent the algorithm from overfitting on one batch. If one or more variants inside the batch are still among the most influential ones out of all

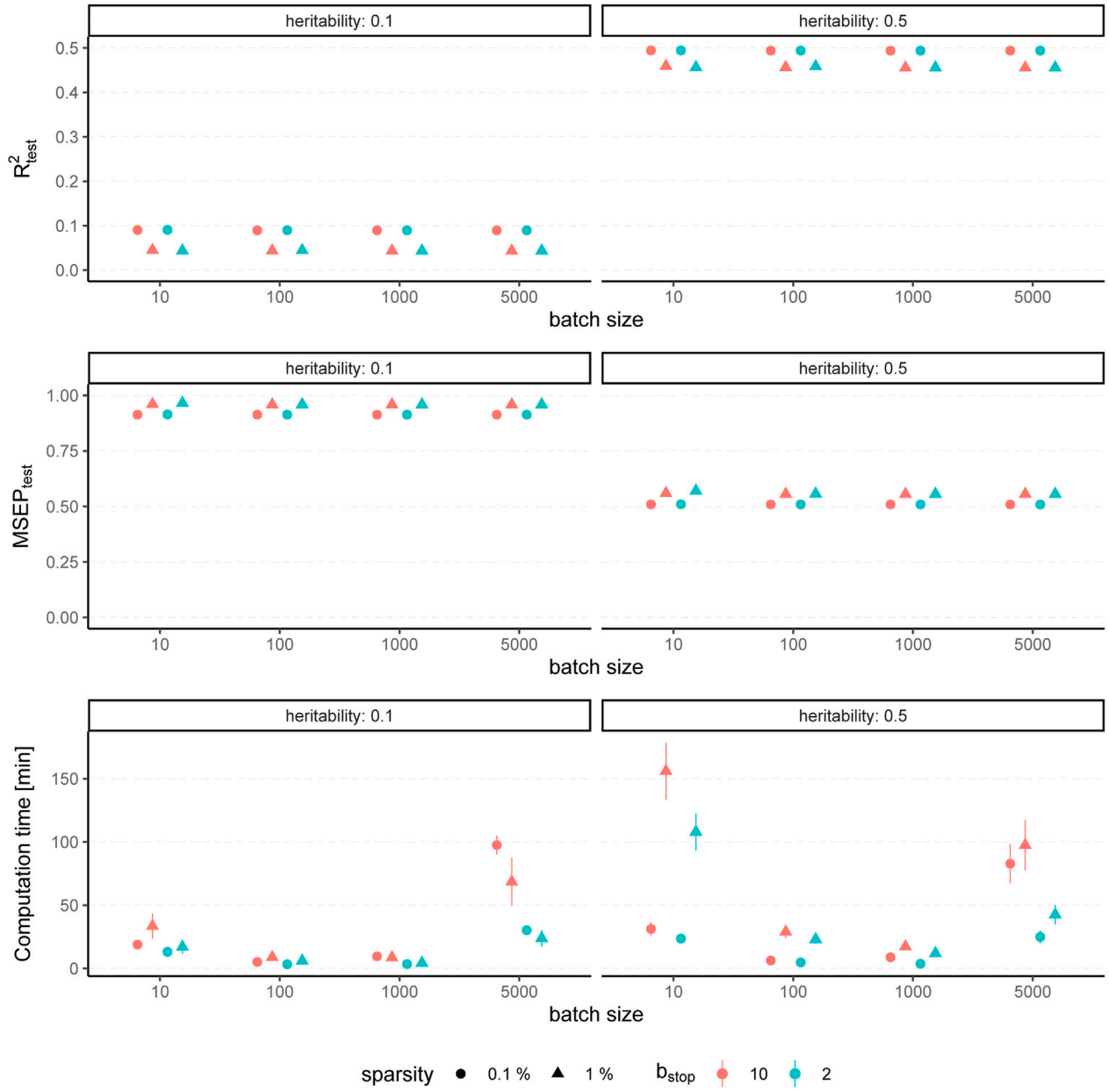

FIGURE 3
Predictive performance for varying batch size and stopping criteria. Results of 100 simulated phenotypes with heritability $h^2 \in \{10\%, 50\%\}$, sparsity $s \in \{0.1\%, 1\%\}$ and $b_{stop} \in \{2, 10\}$ for p = 100, 000 variants and n = 100, 000 individuals (divided into 50% training, 20% validation and 30% test set). Mean and standard deviation of the evaluation metrics are shown depending on the batch size.

variants they will also be included in the next batch. For the outer stopping criterion we specified a large maximum number of batches b_{max} = 20,000 to ensure that the algorithm terminates even in case the MSEP on the validation set has not decreased for b_{stop} batches. Since we do not want the algorithm to stop too early and simultaneously minimize the computation time, in our simulations we consider the choices b_{stop} = 2 and b_{stop} = 10. We then fitted PRS models using snpboost with the previously described hyperparameters. For the computations we used 10 CPUs with 2 GB RAM each.

The results for simulated phenotypes with 10% and 50% heritability are shown in Figure 3 and Figure 4. Results for further degrees of heritability can be found in the supplement. Independently of the heritability and the sparsity of the simulated data, the predictive performance was not affected in our settings by varying batch sizes in terms of R^2 and MSEP. However, the computation time differed crucially, resulting in considerably higher values for rather small (p_{batch} = 10) or rather large (p_{batch} = 5,000) batches. Furthermore, larger batches led to a higher number of included variants in the final model. This effect was stronger for phenotypes which have a less sparse genetic architecture and associated with a later stopping of the algorithm, i.e., more boosting steps were required to derive the final model. A higher number of variants in the final model was associated with a slightly higher MSE of the coefficients as well as higher true positive rates on the one hand but also smaller precision on the other

10.3389/fgene.2022.1076440

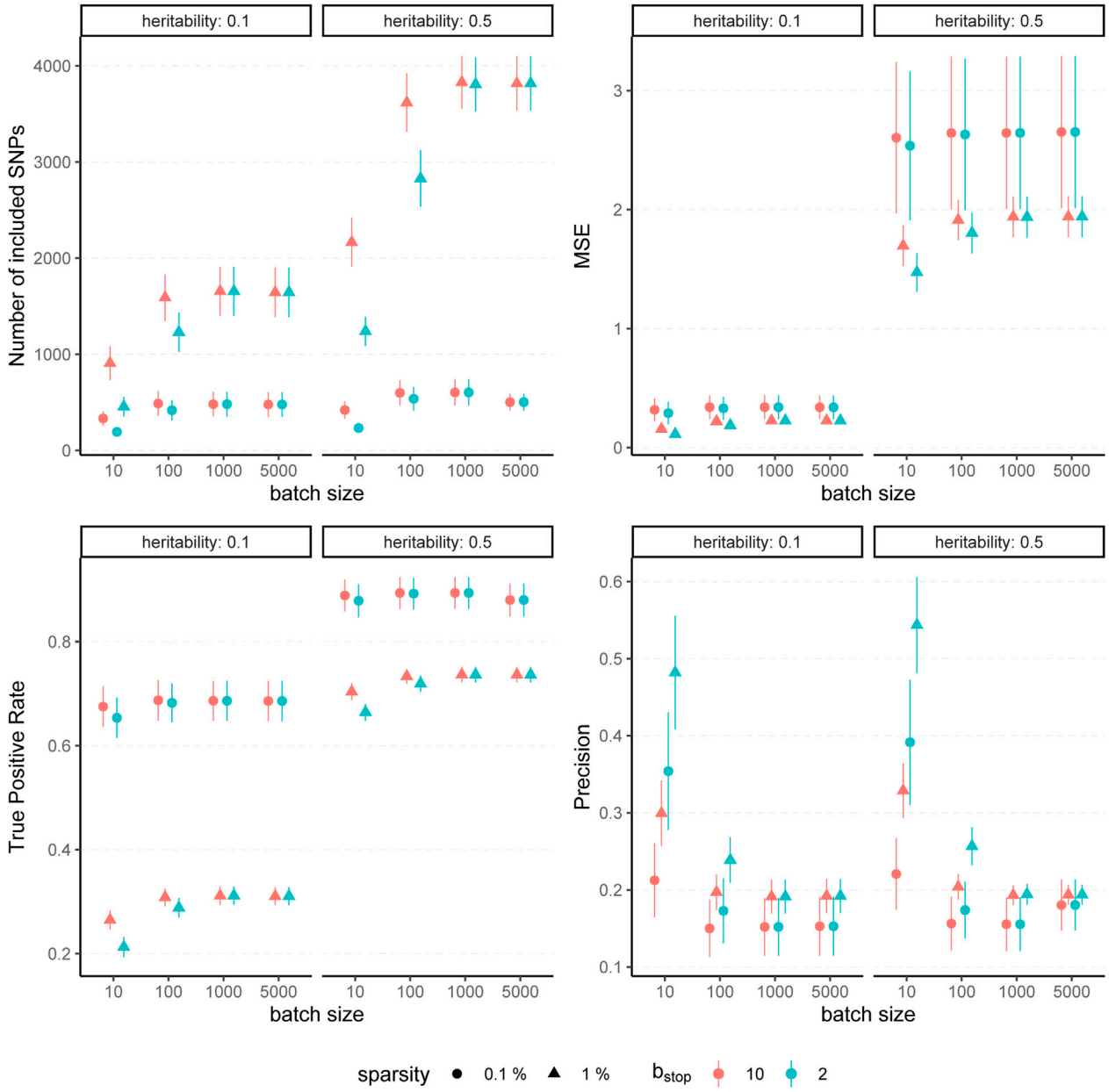

FIGURE 4
Evaluation metrics of the estimated coefficients for varying batch size and stopping criteria. Results of 100 simulated phenotypes with heritability $h^2 \in \{10\%, 50\%\}$, sparsity $s \in \{0.1\%, 1\%\}$ and $b_{stop} \in \{2, 10\}$ for $p = 100{,}000$ variants and $n = 100{,}000$ individuals (divided into 50% training, 20% validation and 30% test set). Mean and standard deviation of the evaluation metrics are shown depending on the batch size.

hand. As expected, a higher m_{stop} increased the computation time of the fitting process for all batch sizes. In contrast, there was no considerable effect on the predictive performance. However, $b_{stop} = 2$ and $b_{stop} = 10$ had an impact on the coefficient estimates as can be seen in Figure 4, e.g., by a tendency to include more variants in the model when choosing $b_{stop} = 10$. This tendency was only apparent for batch sizes $p_{batch} < 1{,}000$, suggesting that for larger batches the choice of b_{stop} is only of minor importance for both, prediction performance and coefficient estimates. The results clearly indicate that a more favorable signal-to-noise ratio (i.e., a higher heritability) and less influential variants (i.e., a higher sparsity) are in general beneficial for the performance of our approach. For phenotypes with a sparser genetic architecture, the considered evaluation metrics tended to show less variability.

In summary, the choice of the hyperparameters had no major influence on the predictive performance measures R^2 and MSEP but on the computation time, which was lowest for medium size batches ($100 \leq p_{batch} \leq 1{,}000$). The accuracy of the coefficient estimates measured *via* MSE, TP and TN rate varied with the batch size, as larger batches tended to lead to more (true positive) variants included in the final model, but also to a slightly higher MSE and a smaller TN rate. While the differences in MSE, TP, and TN rate were only small, smaller batches yielded sparser models in particular for phenotypes with a high heritability.

To conclude, batch sizes of $100 \leq p_{\text{batch}} \leq 1{,}000$ seem to be the most favorable regarding the computation time and the other evaluation metrics. We propose a batch size of $p_{\text{batch}} = 1{,}000$ as the default value because the results suggest less dependency on the b_{stop} parameter than for a batch size of 100 variants. Accordingly, we recommend a default value of $b_{\text{stop}} = 2$ to keep the computation time as low as possible. In practice, genotype data most often contain more than 100,000 variants, which further supports the choice of $p_{\text{batch}} = 1{,}000$ with regard to the computation time. Although our simulation study suggests that those default values should provide reasonable results in most cases, it is recommendable to take the genetic architecture of the examined phenotype as well as the main aim of the analysis into account. Phenotypes with a high expected heritability might be better fitted by using smaller batches, while for phenotypes with many causal variants larger batches might be favorable to increase the TP rate. If one is interested in extremely sparse models identifying only the most-informative variants one could also try to use smaller batches to avoid an overestimation of the number of causal variants.

3.2 Application to the UK Biobank

We applied our proposed method on data from the UK Biobank resource under Application Number 81202. Besides the validation of the results from the previous section, we compared our boosting models fitted *via* the proposed snpboost approach to the ones derived by fitting the lasso *via* the BASIL algorithm implemented in the snpnet package, which have already been shown to outperform commonly-used PRS models for various traits (Qian et al., 2020). Furthermore, we compared our results to PRScs (Ge et al., 2019), LDpred2 (Privé et al., 2021) and SBayesR (Lloyd-Jones et al., 2019), which are based on summary statistics, as well as to multivariable methods *via* LDAK (Zhang et al., 2021) based on Bolt-LMM (Loh et al., 2015), Ridge Regression (Henderson, 1950) and BayesR (Moser et al., 2015).The UK Biobank (UKBB, Bycroft et al., 2018) is a large-scale prospective cohort study including more than half a million participants from the United Kingdom aged between 40 and 69 years when recruited. The database comprises genome-wide genotype data of each individual as well as various in-depth phenotypic information such as biological measurements as well as blood and urine biomarkers. The data have been collected since 2006 and are continually updated with follow-up data.

Our aim is to estimate PRS for various phenotypes, covering several heritability and sparsity levels. The heritability of a trait is an upper bound for the predictive performance based on genotype information. Thus, we used the analyses of Tanigawa et al. (2022) as a proxy and specifically considered five appropriate continuous phenotypes: standing height in cm (UKBB field 50), LDL-cholesterol in mmol/l (UKBB field 30780), blood glucose level in mmol/L (UKBB field 30740), lipoprotein A in nmol/L (UKBB field 30790) and BMI in kg/m^2 (UKBB field 21001).

Height and BMI are quantitative traits with a relatively high heritability and a rather polygenic structure. Twin-studies estimated a heritability of approximately 69% for height and 42% for BMI after adjusting for covariates (Hemani et al., 2013). For a long time, genetic models could not explain this estimated heritability, a phenomenon known as "missing heritability" (Maher, 2008; Gibson, 2010). More recent studies have indicated that this may be primarily due to many influential common variants with small effect sizes (Yang et al., 2010; Wood et al., 2014; Yang et al., 2015) underlining the high polygenicity of those traits. In contrast to this, the distribution of the biomarker lipoprotein A, which is a strong risk factor for coronary heart disease, is mainly explained by variants in the LPA gene on chromosome 6 (Kronenberg and Utermann, 2013). Thus, we expect a sparse PRS with a relatively high prediction accuracy for this trait. For LDL-cholesterol it is known that it is associated with several genes such as LDLR and PCSK9 (Sanna et al., 2011; Sabatine, 2019). Therefore, we expect signal in several genomic regions. Recent studies compared different approaches including the lasso to derive PRS, and found that multivariable methods can reach a predictive performance of up to 20% (Maj et al., 2022; Tanigawa et al., 2022). As in previous works (Sinnott-Armstrong et al., 2021), we adjusted the measured LDL-cholesterol value by a factor of 0.684 for individuals taking statins lowering the blood lipid. For blood glucose we are not aware of a genetic impact and also Tanigawa et al. (2022) found the genetic background only explaining a small fraction (less than 2%) of the biomarker's variance.

Out of the over 500,000 individuals from UK Biobank we filtered for unrelated (based on UKBB resource 668) individuals with self-reported white British ancestry (UKBB field 21000) and available data for all chosen phenotypes, resulting in $n = 262{,}171$ observations. Additionally, the covariates age and sex as well as the first ten principal components of the genotype matrix are available. We randomly divided the data set into training ($n_{\text{train}} = 157{,}204$), validation ($n_{\text{val}} = 52{,}416$) and test set ($n_{\text{test}} = 52{,}551$). We used genome-wide genotype data and filtered for variants with a genotyped rate of at least 90% and a minor allele frequency of at least 0.1%, resulting in $p = 562{,}723$ genetic variants. Missing genotypes are imputed by the corresponding mean of the complete observations.

For both the boosting and lasso approaches, we first estimated a PRS using only the genotyped variants as predictors. We used the training set to fit the model and the validation set to simultaneously monitor the predictive performance for choosing the main tuning parameters of the algorithms (i.e., the number of iterations for boosting and the penalty parameter for the lasso). To fit the lasso we used the R package snpnet (Qian et al., 2020) with the provided default hyperparameters. Following the results of our simulation study, for the snpboost algorithm we chose a batch size of $p_{\text{batch}} = 1{,}000$ variants, a learning rate of $\nu = 0.1$ and an outer stopping lag of $b_{\text{stop}} = 2$ batches. Using the resulting estimated $\widehat{\text{PRS}}$ we fitted two linear models on the combined training and validation set, namely the first one (M_{PRS}) incorporating only the PRS as a single predictor variable:

$$M_{\text{PRS}}\colon Y = \gamma_0 + \gamma_{\text{PRS}}\widehat{\text{PRS}} \tag{4}$$

and the second one (M_f) including the first ten principal components, sex and age as additional covariates:

$$M_f\colon Y = \gamma_0 + \gamma_{\text{PRS}}\widehat{\text{PRS}} + \gamma_1\text{PC}_1 + \cdots + \gamma_{10}\text{PC}_{10} + \gamma_{\text{sex}}\text{sex} + \gamma_{\text{age}}\text{age}. \tag{5}$$

To measure the actual benefit in accuracy of including a PRS in the prediction model, we also fitted a model including only covariates (M_c):

$$M_c\colon Y = \gamma_0 + \gamma_1\text{PC}_1 + \cdots + \gamma_{10}\text{PC}_{10} + \gamma_{\text{sex}}\text{sex} + \gamma_{\text{age}}\text{age}. \tag{6}$$

Finally, we also included the covariates in the fitting process to derive the PRS, corresponding to the model $M_{PRS,c}$:

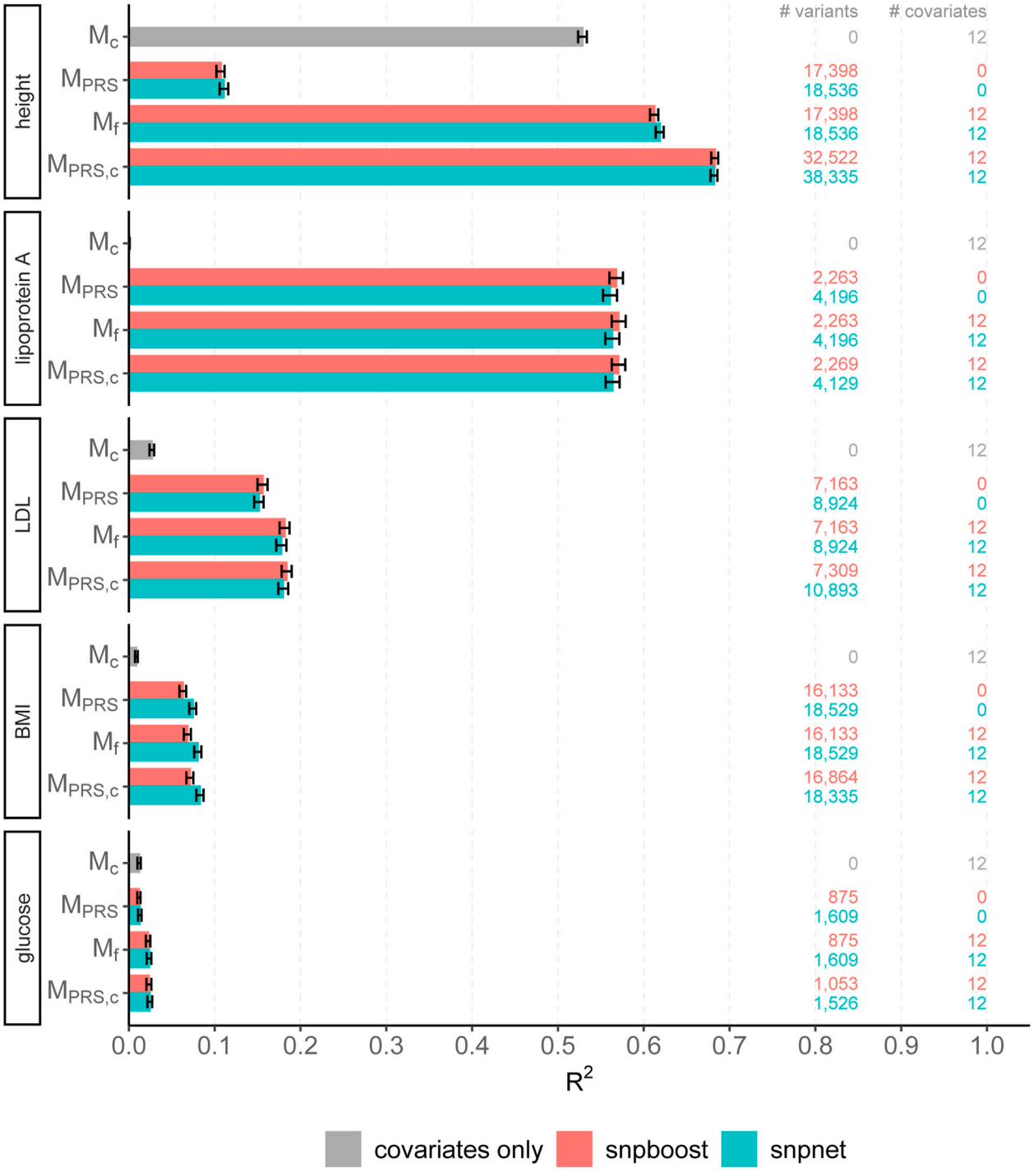

FIGURE 5
Comparison of predictive performance of snpnet and snpboost for five continuous phenotypes from the UKBB. Results of the covariate-only model (M_c; grey bars) and multivariable polygenic models with and without inclusion of the covariates derived by lasso (snpnet; petrol-colored bars) and statistical boosting (snpboost; red bars) for the prediction of five phenotypes from the UKBB. The barplots show the predictive performance (R^2) on the test set of 52,551 unrelated white British individuals. M_{PRS} corresponds to a linear model incorporating the PRS as a single predictor variable and M_f to a linear model incorporation sex, age and the first ten principal components as additional covariates. $M_{PRS,c}$ includes the covariates already in the fitting process of the PRS. Bootstrapped 95% confidence intervals are indicated by error bars. Furthermore, information on the number of selected genetic variants (# variants) and the number of additionally included covariates (# covariates) is given.

TABLE 2 Comparison of computational efficiency of snpnet and snpboost on eight phenotypes from the UKBB. Computational times of the algorithms snpnet and snpboost for multivariable polygenic models with and without inclusion of the covariates for the prediction of eight phenotypes from the UKBB. M_{PRS} corresponds to the application of the algorithms without including covariates and $M_{PRS,c}$ to the inclusion of the covariates sex, age and the first ten principal components. The experiments were run on 16 CPUs with 2 GB RAM each.

		Computation time in minutes	
Phenotype	Model	snpnet	snpboost
Height	M_{PRS}	132.44	116.65
Height	$M_{PRS,c}$	97.49	299.98
BMI	M_{PRS}	54.36	94.34
BMI	$M_{PRS,c}$	54.81	156.49
LDL	M_{PRS}	37.92	50.64
LDL	$M_{PRS,c}$	45.61	64.27
glucose	M_{PRS}	14.86	11.38
glucose	$M_{PRS,c}$	14.71	16.33
lipoprotein A	M_{PRS}	28.99	25.08
lipoprotein A	$M_{PRS,c}$	33.67	30.14
asthma	M_{PRS}	3.97	5.31
asthma	$M_{PRS,c}$	4.21	6.63
coeliac	M_{PRS}	3.11	5.46
coeliac	$M_{PRS,c}$	5.00	6.69
HBP	M_{PRS}	46.63	90.27
HBP	$M_{PRS,c}$	30.07	181.23

$$M_{PRS,c}: Y = \beta_0 + \beta_{PC1}\mathrm{PC}_1 + \cdots + \beta_{PC10}\mathrm{PC}_{10} + \beta_{sex}\mathrm{sex} + \beta_{age}\mathrm{age} + \sum_{j=1}^{P} \beta_j X_j. \quad (7)$$

All models were evaluated on the independent test set and compared with respect to their predictive performance, computational efficiency and sparsity. To measure the predictive performance we used the R^2 value on the test set given by the squared correlation between the observed and predicted phenotypes as well as the root mean squared error of prediction (RMSEP). The computational efficiency was measured as the computation time in minutes of the respective algorithm and the sparsity is given by the number of included variants in the final PRS. All computations were conducted on a computer cluster with 16 CPUs and 2 GB RAM each. The derivation of the PRS by the use of further methods (namely PRScs, LDpred2, SBayesR, Bolt-LMM, Ridge Regression and BayesR) was based on the same training and validation data and is described in the Supplementary Material. All models were tested on the same independent test set.

The results of snpboost as well as of snpnet for all phenotypes are given in Figure 5 and Table 2. The resulting RMSEP is shown in Supplementary Figure S7. Overall, snpnet and snpboost yield comparable results regarding the predictive performance, without one approach being consistently superior to the other. Both the resulting R^2 and RMSEP are very close. Furthermore, the shown R^2 values are in line with previously reported R^2 resulting from snpnet, which has been shown to be highly competitive to various other (univariate) PRS methods (Qian et al., 2020; Li et al., 2022; Tanigawa et al., 2022). The PRS estimated *via* snpnet and snpboost both clearly increase the predictive performance compared to the covariates-only model M_c for all shown phenotypes. With respect to sparsity, our boosting approach tends to select less variants (on average 26% less variants compared to the lasso). The computation time of both approaches is highly dependent on the genetic architecture, i.e., the heritability and sparsity of the phenotype. In particular, a higher and more polygenic signal tends to lead to longer computation times. In case of fitting the PRS based solely on the genotype data and including the covariates in a subsequent linear model, snpboost tends to be faster than snpnet; however, the computation times for snpboost increase substantially when including covariates in the fitting process for LDL-cholesterol and height. This is partly due to more coefficients being fitted and updated in each boosting step and partly due to larger PRS models resulting from more boosting steps. Nevertheless, the models are still fitted in reasonable time using our batch-based approach. As described in Hepp et al. (2016), boosting is generally expected to be slower than the lasso, which can only be observed for less sparse models in the examined phenotypes. In general, the model $M_{PRS,c}$ outperforms the model M_f regarding the predictive performance, implying that including the covariates already in the fitting of the PRS is favorable regarding the detection of the genetic signal. However, the effect is only substantial for phenotypes with a high association to covariates (i.e., height). Furthermore, the model $M_{PRS,c}$ tends to select more variables than estimating the PRS based only on the genotypes (M_{PRS}) and the computation time is considerably increased when using the snpboost approach. Therefore, it might be advisable to only consider the covariates in the fitting process if there is a large association already in the covariates-only model.

Figure 6 displays the absolute values of the resulting estimated non-zero coefficients for LDL-cholesterol for the boosting and lasso approaches. Both tend to detect variants with higher effect sizes in the same genetic regions, e.g., at chromosome 2 and chromosome 19. In total, there are 3,030 genetic variants that are present in both PRS, out of 7,163 variants selected by snpboost and 8,924 variants selected by snpnet. While snpboost results in less variants, the included variants have larger effect sizes and less variants with very small effect sizes close to zero are included in the model. Supplementary Figure S8 displays the coefficients again with shared variants marked in black. All SNPs with comparably high effect sizes in the snpnet PRS are included in both models but the snpboost PRS incorporates further SNPs with stronger effects. The results are similar for the other phenotypes and included in the Supplementary Material. In conclusion, the snpboost PRS tends to include less variants in total, but more variants with comparably high effect sizes corresponding to less shrinkage for the variants included in the model compared to the lasso.

The Supplementary Material comprises results for comparisons to further commonly used methods to derive PRS (Supplementary Tables S1, S2). Our proposed algorithm yielded consistently higher prediction performance compared to the summary statistics based PRScs and LDpred2 methods; furthermore, it yielded competitive results compared to summary statistics based SBayesR and four different multivariable approaches, while tending to select the sparsest models.

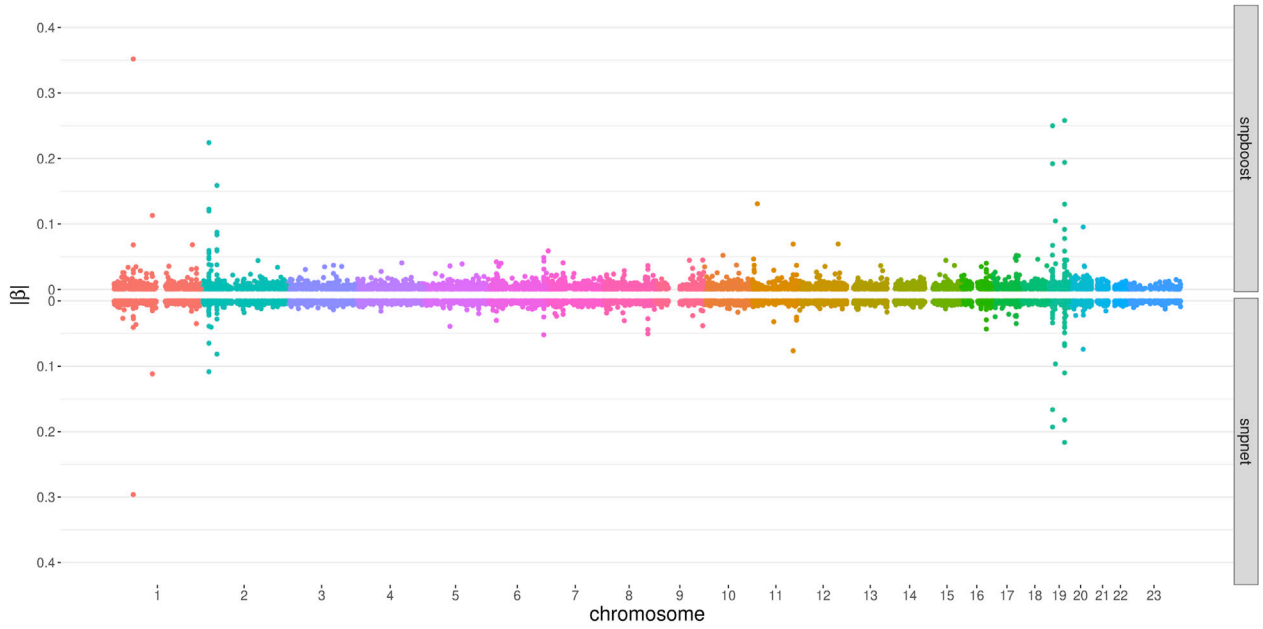

FIGURE 6
Absolute values of coefficient estimates for PRS models for LDL-cholesterol derived by boosting (snpboost) and lasso (snpnet) in dependence of the genomic position of the variants.

4 Extension to binary data

While traits like blood biomarkers or physical measurements are often quantitative, it is, for example, also of interest to predict the probability of the occurrence of a disease for a particular patient. In this case we deal with binary data $y_i \in \{0, 1\}$ and proceed as in a logistic regression by modelling the logit of the expected value as a linear model

$$\text{logit}\left(\mathrm{P}\left(y_i = 1|\mathbf{X}\right)\right) = \ln\left(\frac{\mathrm{P}\left(y_i = 1|\mathbf{X}\right)}{1 - \mathrm{P}\left(y_i = 1|\mathbf{X}\right)}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}, \quad i = 1, \ldots, n. \tag{8}$$

The estimated probability $\widehat{p}_i\left(\mathbf{X}\right) = \mathrm{P}\left(y_i = 1|\mathbf{X}\right)$ is then given by

$$\widehat{p}_i\left(X\right) = \mathrm{P}\left(y_i = 1|\mathbf{X}\right) = \frac{\exp\left(\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{i,j}\right)}{1 + \exp\left(\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{i,j}\right)}. \tag{9}$$

To fit binary outcomes *via* boosting we replace the L_2 loss by the log loss

$$f_{\ln}\left(\mathbf{y}, \hat{\mathbf{p}}\right) = -\frac{1}{n} \sum_{i=1}^{n} y_i \ln\left(\hat{p}_i\right) + \left(1 - y_i\right) \ln\left(1 - \hat{p}_i\right). \tag{10}$$

Note that following this new loss function, the gradient is no longer represented by the residuals. The base-learners are hence fitted now to the first derivative of the loss function in Eq. 10. Consequently, batches are built out of the p_batch variants with the highest absolute correlation to the first derivative of the loss in Eq. 10 instead of the residual. However, the other components of the algorithm including the base learners remain unchanged. We also keep the hyperparameters derived in Section 3.1.2 fixed. We applied the extended algorithm on data of the UKBB for three binary phenotypes: the occurrence of asthma (UKBB field 22127), coeliac disease (UKBB field 21068) and high blood pressure (UKBB field 6150). All three traits are associated to many environmental factors but also have a genetic component (Arora and Newton-Cheh, 2010; Trynka et al., 2010; Yang et al., 2017; El-Husseini et al., 2020). Tanigawa et al. (2022) estimated high blood pressure to be a rather polygenic trait while the genetic component of asthma and coeliac disease is determined by fewer common variants.

We incorporated unrelated individuals of white British ancestry in our analysis and divided the samples randomly into training, validation and test sets. In total we used 8,397 cases (n_{train} = 4,266, n_{val} = 1,709 and n_{test} = 2,522) and 58,428 controls (n_{train} = 29,079, n_{val} = 11,707, n_{test} = 17,642) for asthma, 1,793 cases (n_{train} = 882, n_{val} = 361 and n_{test} = 550) and 92,646 controls (n_{train} = 46,234, n_{val} = 18,449, n_{test} = 27,963) for coeliac disease and 71,235 cases (n_{train} = 35,720, n_{val} = 14,210 and n_{test} = 21,305) and 190,422 controls (n_{train} = 94,740, n_{val} = 38,246, n_{test} = 57,436) for high blood pressure.

The applicability to binary traits was also one of the first extension of snpnet and Qian et al. (2020) showed impressive results for a number of binary traits. Due to that, we again also apply snpnet to the same data to evaluate the quality of our results.

We evaluated the accuracy of the resulting predictions on the test set using both the log loss as well as the AUC. Results are shown in Figure 7 and in Supplementary Figure S17. The overall predictive performance is comparable for all three phenotypes. For high blood pressure with a polygenic genetic component snpboost yields a sparse model with a high predictive performance. For sparse binary phenotypes as asthma and coeliac disease, snpboost and snpnet yield similar sparse models. The result for coeliac disease, which appears to be rather oliogenic than polygenic, for snpnet is outstanding, but in line with the results of Tanigawa et al. (2022). Nevertheless, also snpboost also estimates a very sparse PRS with a

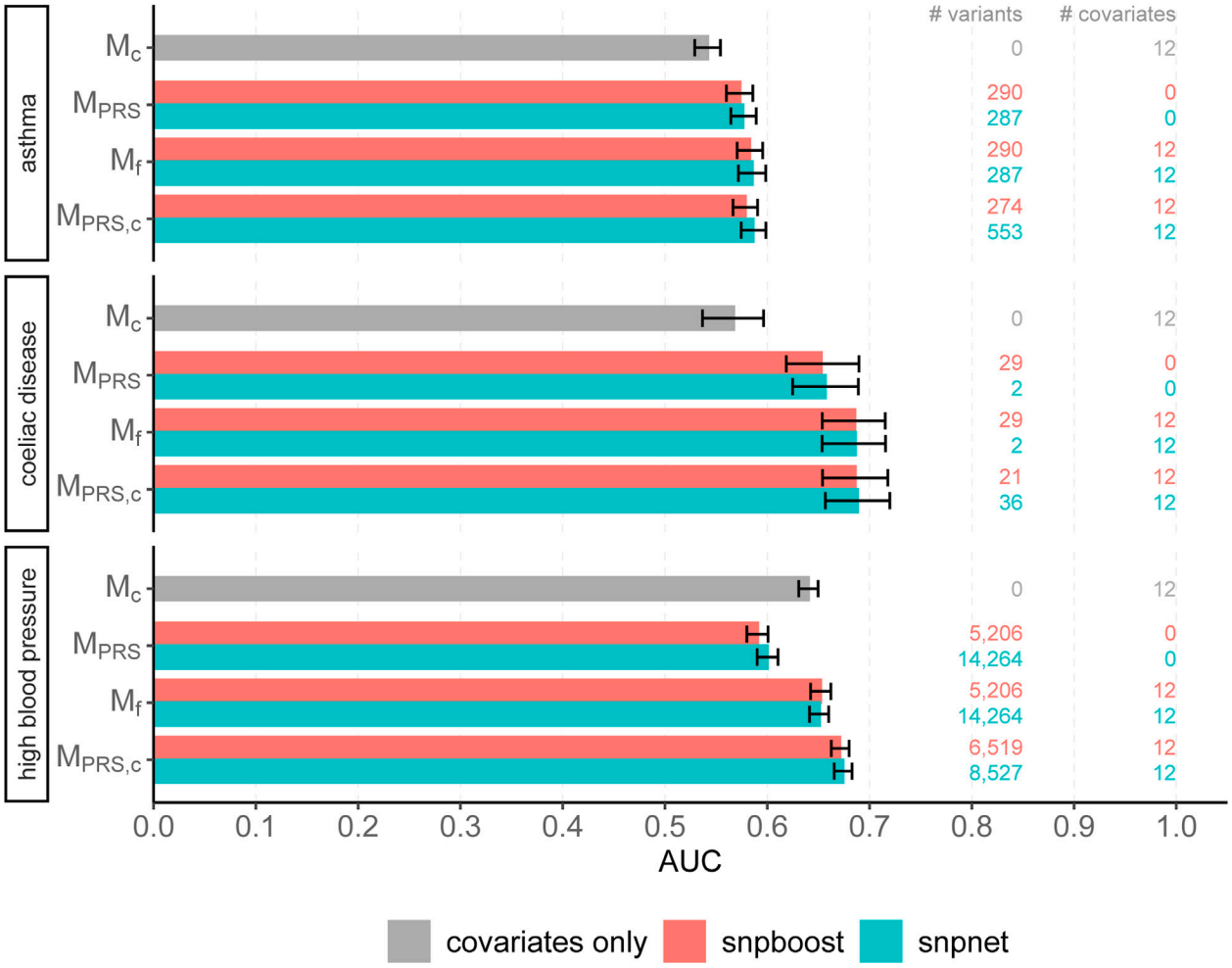

FIGURE 7
Comparison of predictive performance of snpnet and snpboost for three binary phenotypes from the UKBB. Results of the covariate-only model (M_c; grey bars) and multivariable polygenic models with and without inclusion of the covariates derived by lasso (snpnet; petrol-coloured bars) and statistical boosting (snpboost; red bars) for the prediction of three binary phenotypes from the UKBB. The barplots show the AUC on the test set of 20,164 (asthma), 28,513 (coeliac disease) and 78,741 (high blood pressure) unrelated white British individuals. M_{PRS} corresponds to a logistic regression model incorporating the PRS as a single predictor variable and M_f to a logistic regression model incorporating sex, age and the first ten principal components as additional covariates. $M_{PRS,c}$ includes the covariates already in the fitting process of the PRS. Bootstrapped 95% confidence intervals are indicated by error bars. Furthermore, information on the number of selected genetic variants (# variants) and the number of additionally included covariates (# covariates) is given.

high predictive performance. Table 2 illustrates the computation time for binary data of snpnet and snpboost on a computer cluster with 16 CPUs and 2 GB RAM each. Both, snpboost and snpnet yield very limited computation times, with snpnet being faster.

In summary, this illustrates how easily and conveniently the snpboost framework can be extended to different data types by incorporating different loss functions. Even though we simply plugged in the log loss and did not optimize the hyperparameters such as the batch size or the learning rate of our algorithm for binary data, snpboost yields a competitive predictive performance compared to the BASIL algorithm.

5 Discussion

In this work we have proposed a new methodological framework to derive multivariable PRS models *via* applying a statistical boosting approach directly on genotype data. Currently, PRS are most often built based on summary statistics from GWAS that were estimated by simple and univariate linear regression models (Choi et al., 2020). This methodologically simple approach is mainly justified by the computational hurdle resulting from the ultra-high-dimensionality of the genotype data. For example, in the past it had been unfeasible to fit a lasso model on the complete genotype data due to the high computational complexity. To overcome this, Mak et al. (2017) developed lassosum, an approach to approximate the lasso path by only using summary statistics and LD reference panels. However, recently published works provided methods to enable statistical modelling by penalized multivariable regression approaches on genotype data (Privé et al., 2018; Qian et al., 2020). Qian et al. demonstrated that lasso-based PRS were able to outperform several PRS derived by methods based on univariate summary statistics (Qian et al., 2020). First approaches to apply statistical boosting on genotype data employed a three-step-approach to fit multivariable PRS (Maj

et al., 2022): first, variants are pre-filtered based on their univariate associations with the examined phenotype. Second, statistical modelling and variable selection approaches such as AdaSub (Staerk et al., 2021) and boosting with probing (Thomas et al., 2017) are used to identify the informative variants on blocks of variants in LD. Finally, a multivariable PRS based on the selected variants is constructed *via* component-wise L_2-boosting (Bühlmann and Hothorn, 2007). While this approach yielded particularly sparse models and could compete with common methods like clumping and thresholding (Euesden et al., 2014), lasso *via* snpnet yielded more accurate results regarding the predictive performance which is usually the main objective of PRS modelling.

In the present article we introduced the boosting algorithm snpboost that works on smaller batches of variants similar to the BASIL algorithm. Our framework enables the application of statistical boosting directly on the complete original genotype data. In a smaller but still high-dimensional simulation setting, we were able to show that the adapted boosting algorithm yields similar performance to the original component-wise L_2-boosting, indicating that we do not lose predictive performance due to the incorporation of batches. In a further setting with more realistic dimensionality we have derived appropriate default values for the application of snpboost on large-scale data. We were able to show that the specified default values resulted in models with good performance in most cases but also gave advice on how to adapt them based on the genetic architecture of the examined phenotype and the specific research questions.

We applied the newly proposed snpboost algorithm on large-scale genotype data of the UKBB. In particular, we have compared the performance of snpboost to the one achieved by the lasso *via* snpnet, which has been shown to outperform many classical PRS (Qian et al., 2020). Our results indicate that the snpboost algorithm leads to PRS models that are highly competitive to lasso-based PRS models in both predictive performance and computation time. Although it might have been expected that the computation time would be higher for statistical boosting than for the lasso (Hepp et al., 2016), our approach had a tendency to result in sparser models. These sparser models correspond to an earlier stopping of the algorithm which reduces the computation time of boosting. The incorporation of further covariates such as age, sex and principal components in the fitting process of the PRS resulted in increased computation times for some phenotypes, particularly for height. However, in such cases, the boosting algorithm yielded an improved predictive performance with larger numbers of included variants. This illustrates that sparsity is not always favorable in regards of predictive performance. Additionally, we compared the performance of snpboost to further predictive PRS tools, which are either summary statistics based as PRScs, LDpred2 and SBayesR or multivariable approaches *via* the LDAK implementation of BayesR, Ridge Regression and Bolt-LMM (Zhang et al., 2021). While these methods do not apply variable selection, the predictive performance of snpboost was still highly competitive.

Our analyses show that there is a large overlap of the chosen variants by lasso and boosting, in particular regarding the variants with high estimated effect sizes. However, boosting has been found to include less variants in the final model and to induce less shrinkage on the effect estimates compared to the lasso. In clinical practice, a sparser PRS model might be of particular interest if the aim is not only prediction but also the identification of risk loci in the genome. In fact, functional annotations of the selected variants can better elucidate the underlying biological mechanisms influencing the analyzed trait. Thus, statistical boosting might be one way to yield more biologically interpretable PRS models.

Despite the presented promising results, the proposed method also inherited some limitations from statistical boosting. In contrast to classical regression methods, boosting does not provide closed formulas for standard errors of effect estimates or confidence intervals that could be used for inference. Furthermore, as mentioned before, statistical boosting is in general associated with a slightly higher computational complexity compared to methods such as the lasso (Hepp et al., 2016) and has a known tendency to include too many variables in low-dimensional settings (Staerk and Mayr, 2021; Strömer et al., 2022). Our results suggest that the incorporation of batches substantially reduced the computational time. Additionally, the reduction of the search space in each boosting step might partially prevent the algorithm from selecting too many variables. However, the implementation of the batch-based statistical boosting in snpboost is currently limited to linear base-learners, each corresponding to one genetic variant.

Apart from those technical limitations, using individual-level data raises ethical and logistical questions: While summary statistics are easily shared and do not allow for identification of unique individuals, individual-level data involve the risk of identification. It is therefore crucial, that providers as well as researchers using individual-level data follow ethical standards. Furthermore, the storage and transfer of individual-level data require more capacities which might not be at everyone's disposal in the complete research community. However, the resulting PRS can be published by sharing only the included variants, alleles and coefficients—exactly like summary statistics (Lambert et al., 2021). To make use of available summary statistics and to avoid the limitations associated with individual-level data, it might be of interest to develop an approximation of a component-wise boosting algorithm based on summary statistics and LD panels, analogously as lassosum for the lasso. From a computational perspective, this is not necessary as snpboost only requires limited resources (e.g., our analysis of the UKBB data was run with only 32 GB RAM in total).

By incorporating the log loss we made our framework applicable also to binary traits and demonstrated the convenience of further extensions of the snpboost framework beyond the case of continuous phenotypes. Without re-specifying our hyperparameters we were able to yield similar results as the snpnet framework.

In future research we want to further exploit the modular structure of boosting to model more complex biological phenomena. We will incorporate different loss functions to extend the snpboost framework to be applicable also to count and time-to-event data. To account for the uncertainty in the prediction, one could also construct subject-specific prediction intervals based on quantile regression (Mayr et al., 2012). Besides extending the approach *via* new loss functions, one could also change the base-learners in various ways. For example, base-learners could be adapted to take into account different models of inheritance beside the classical additive component typically used in the polygenic models, such as dominant and recessive hereditary schemes. Further possibilities for future research include the extension of the set of possible base-learners, e.g., to model gene-environment interactions as well as epistatic effects across variants which can play a relevant role in biological phenotypes (Li and Lehner, 2020). To do so, base-learners including interactions between variants and variant-covariate interactions could be incorporated. Apart from that, biological knowledge can also be used a priori. Márquez-Luna et al. (2021) have shown that the incorporation of functional annotations of the genetic variants contribute to a rise in

prediction accuracy. Previous works in the field of penalized regression and boosting have proposed to make use of biologically meaningful groups of genomic variants such as genes or pathways as described by Luan and Li (2008), Wei and Li (2007) as well as Liu et al. (2013). While those previous methods were computationally limited to smaller datasets our framework opens the possibility to include those ideas in the multivariable modelling of PRS. Besides those methodological extensions of our proposed snpboost framework, future research will also focus on the practical application of PRS derived by our framework. An important aspect of PRS research is the transferability of PRS models to different ethnicities, as PRS are often derived on cohorts of European ancestry and a substantial loss of predictive performance is observed when applied on further cohorts with different ethnicities (Landry et al., 2018; Evans et al., 2022). Previous studies have indicated that sparser models may contribute to overcome this issue (Maj et al., 2022) and it is of particular interest to examine the transferability of PRS derived by statistical boosting.

Data availability statement

The data analyzed in this study is subject to the following licenses/restrictions: This research has been conducted using the UK Biobank resource under application number 81202 (http://www.ukbiobank.ac.uk). Requests to access these datasets should be directed to UK Biobank, http://www.ukbiobank.ac.uk.

Ethics statement

The studies involving human participants were reviewed and approved by the North West Multi-centre Research Ethics Committee (MREC) as a Research Tissue Bank (RTB) approval. (UK Biobank, https://www.ukbiobank.ac.uk/learn-more-about-uk-biobank/about-us/ethics). The patients/participants provided their written informed consent to participate in this study.

Author contributions

HK, AM, CS, CM, and PK contributed to conception and design of the method. HK wrote the code, performed the experiments and wrote the first draft of the manuscript. All authors contributed to manuscript revision, read and approved the submitted version.

Funding

The work on this article was supported by the Deutsche Forschungsgemeinschaft (DFG, grant number 428239776, MA7304/1-1).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2022.1076440/full#supplementary-material

References

Arora, P., and Newton-Cheh, C. (2010). Blood pressure and human genetic variation in the general population. *Curr. Opin. Cardiol.* 25, 229–237. doi:10.1097/hco.0b013e3283383e2c

Beesley, L. J., Salvatore, M., Fritsche, L. G., Pandit, A., Rao, A., Brummett, C., et al. (2020). The emerging landscape of health research based on biobanks linked to electronic health records: Existing resources, statistical challenges, and potential opportunities. *Statistics Med.* 39, 773–800. doi:10.1002/sim.8445

Bühlmann, P., and Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Stat. Sci.* 22, 477–505. doi:10.1214/07-STS242

Bühlmann, P., and Yu, B. (2003). Boosting with the l2 loss. *J. Am. Stat. Assoc.* 98, 324–339. doi:10.1198/016214503000125

Bunea, F., Tsybakov, A., and Wegkamp, M. (2007). Sparsity oracle inequalities for the Lasso. *Electron. J. Statistics* 1, 169–194. doi:10.1214/07-EJS008

Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L. T., Sharp, K., et al. (2018). The UK Biobank resource with deep phenotyping and genomic data. *Nature* 562, 203–209. doi:10.1038/s41586-018-0579-z

Chang, C., Chow, C. C., Tellier, L. C., Vattikuti, S., Purcell, S. M., and Lee, J. J. (2015). Second-generation PLINK: Rising to the challenge of larger and richer datasets. *GigaScience* 4, 7. doi:10.1186/s13742-015-0047-8

Chen, T., and Guestrin, C. (2016). "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (New York, NY, USA: Association for Computing Machinery), KDD '16), 785. doi:10.1145/2939672.2939785

Choi, S. W., Mak, T. S.-H., and O'Reilly, P. F. (2020). Tutorial: A guide to performing polygenic risk score analyses. *Nat. Protoc.* 15, 2759–2772. doi:10.1038/s41596-020-0353-1

El-Husseini, Z. W., Gosens, R., Dekker, F., and Koppelman, G. H. (2020). The genetics of asthma and the promise of genomics-guided drug target discovery. *Lancet Respir. Med.* 8, 1045–1056. doi:10.1016/s2213-2600(20)30363-5

Euesden, J., Lewis, C. M., and O'Reilly, P. F. (2014). PRSice: Polygenic risk score software. *Bioinformatics* 31, 1466–1468. doi:10.1093/bioinformatics/btu848

Evans, D. G., van Veen, E. M., Byers, H., Roberts, E., Howell, A., Howell, S. J., et al. (2022). The importance of ethnicity: Are breast cancer polygenic risk scores ready for women who are not of white European origin? *Int. J. Cancer* 150, 73–79. doi:10.1002/ijc.33782

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statistics* 29, 1189–1232. doi:10.1214/aos/1013203451

Fu, W., and Knight, K. (2000). Asymptotics for lasso-type estimators. *Ann. Statistics* 28, 1356–1378. doi:10.1214/aos/1015957397

Ge, T., Chen, C.-Y., Ni, Y., Feng, Y.-C. A., and Smoller, J. W. (2019). Polygenic prediction via bayesian regression and continuous shrinkage priors. *Nat. Commun.* 10, 1776. doi:10.1038/s41467-019-09718-5

Gibson, G. (2010). Hints of hidden heritability in GWAS. *Nat. Genet.* 42, 558–560. doi:10.1038/ng0710-558

Greenshtein, E., and Ritov, Y. (2004). Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli* 10, 971–988. doi:10.3150/bj/1106314846

Hassanin, E., May, P., Aldisi, R., Spier, I., Forstner, A. J., Nöthen, M. M., et al. (2021). Breast and prostate cancer risk: The interplay of polygenic risk, rare pathogenic germline variants, and family history. *Genet. Med.* 24, 576–585. doi:10.1016/j.gim.2021.11.009

Hemani, G., Yang, J., Vinkhuyzen, A., Powell, J., Willemsen, G., Hottenga, J.-J., et al. (2013). Inference of the genetic architecture underlying bmi and height with the use of 20, 240 sibling pairs. *Am. J. Hum. Genet.* 93, 865–875. doi:10.1016/j.ajhg.2013.10.005

Henderson, C. R. (1950). Estimation of genetic parameters. *Ann. Math. Stud.* 21, 309.

Hepp, T., Schmid, M., Gefeller, O., Waldmann, E., and Mayr, A. (2016). Approaches to regularized regression – a comparison between gradient boosting and the lasso. *Methods Inf. Med.* 55, 422–430. doi:10.3414/ME16-01-0033

Hoerl, A. E., and Kennard, R. W. (2000). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 42, 80–86. doi:10.1080/00401706.2000.10485983

Hothorn, T., Buehlmann, P., Kneib, T., Schmid, M., and Hofner, B. (2010). Model-based Boosting 2.0. *J. Mach. Learn. Res.* 11, 2109–2113.

National Human Genome Research Institute (2021). The cost of sequencing a human genome. Available at: https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost (Accessed 11 04, 2021).

Kronenberg, F., and Utermann, G. (2013). Lipoprotein(a): Resurrected by genetics. *J. Intern. Med.* 273, 6–30. doi:10.1111/j.1365-2796.2012.02592.x

Kuchenbaecker, K. B., Hopper, J. L., Barnes, D. R., Phillips, K.-A., Mooij, T. M., Roos-Blom, M.-J., et al. (2017). Risks of breast, ovarian, and contralateral breast cancer for brca1 and brca2 mutation carriers. *JAMA* 317, 2402–2416. doi:10.1001/jama.2017.7112

Lambert, S. A., Gil, L., Jupp, S., Ritchie, S. C., Xu, Y., Buniello, A., et al. (2021). The polygenic score catalog as an open database for reproducibility and systematic evaluation. *Nat. Genet.* 53, 420–425. doi:10.1038/s41588-021-00783-5

Landry, L. G., Ali, N., Williams, D. R., Rehm, H. L., and Bonham, V. L. (2018). Lack of diversity in genomic databases is a barrier to translating precision medicine research into practice. *Health Aff.* 37, 780–785. doi:10.1377/hlthaff.2017.1595

Li, R., Chang, C., Justesen, J. M., Tanigawa, Y., Qian, J., Hastie, T., et al. (2022). Fast Lasso method for large-scale and ultrahigh-dimensional Cox model with applications to UK Biobank. *Biostatistics* 23, 522–540. doi:10.1093/biostatistics/kxaa038

Li, X., and Lehner, B. (2020). Biophysical ambiguities prevent accurate genetic prediction. *Nat. Commun.* 11, 4923. doi:10.1038/s41467-020-18694-0

Liu, J., Huang, J., Ma, S., and Wang, K. (2013). Incorporating group correlations in genome-wide association studies using smoothed group lasso. *Biostatistics* 14, 205–219. doi:10.1093/biostatistics/kxs034

Lloyd-Jones, L. R., Zeng, J., Sidorenko, J., Yengo, L., Moser, G., Kemper, K. E., et al. (2019). Improved polygenic prediction by bayesian multiple regression on summary statistics. *Nat. Commun.* 10, 5086. doi:10.1038/s41467-019-12653-0

Loh, P.-R., Tucker, G., Bulik-Sullivan, B. K., Vilhjálmsson, B. J., Finucane, H. K., Salem, R. M., et al. (2015). Efficient bayesian mixed-model analysis increases association power in large cohorts. *Nat. Genet.* 47, 284–290. doi:10.1038/ng.3190

Luan, Y., and Li, H. (2008). Group additive regression models for genomic data analysis. *Biostatistics* 9, 100–113. doi:10.1093/biostatistics/kxm015

Maher, B. (2008). Personal genomes: The case of the missing heritability. *Nature* 456, 18–21. doi:10.1038/456018a

Maj, C., Staerk, C., Borisov, O., Klinkhammer, H., Yeung, M. W., Krawitz, P., et al. (2022). Statistical learning for sparser fine-mapped polygenic models: The prediction of LDL-cholesterol. *Genet. Epidemiol.* 46, 589–603. doi:10.1002/gepi.22495

Mak, T. S. H., Porsch, R. M., Choi, S. W., Zhou, X., and Sham, P. C. (2017). Polygenic scores via penalized regression on summary statistics. *Genet. Epidemiol.* 41, 469–480. doi:10.1002/gepi.22050

Márquez-Luna, C., Gazal, S., Loh, P.-R., Kim, S. S., Furlotte, N., Auton, A., et al. (2021). Incorporating functional priors improves polygenic prediction accuracy in UK biobank and 23andme data sets. *Nat. Commun.* 12, 6052. doi:10.1038/s41467-021-25171-9

Mayr, A., and Hofner, B. (2018). Boosting for statistical modelling-a non-technical introduction. *Stat. Model.* 18, 365–384. doi:10.1177/1471082X17748086

Mayr, A., Hothorn, T., and Fenske, N. (2012). Prediction intervals for future BMI values of individual children - a non-parametric approach by quantile boosting. *BMC Med. Res. Methodol.* 12, 6. doi:10.1186/1471-2288-12-6

Meinshausen, N. (2007). Relaxed lasso. *Comput. Statistics Data Analysis* 52, 374–393. doi:10.1016/j.csda.2006.12.019

Moser, G., Lee, S. H., Hayes, B. J., Goddard, M. E., Wray, N. R., and Visscher, P. M. (2015). Simultaneous discovery, estimation and prediction analysis of complex traits using a bayesian mixture model. *PLOS Genet.* 11, e1004969. doi:10.1371/journal.pgen.1004969

Privé, F., Arbel, J., and Vilhjálmsson, B. J. (2021). Ldpred2: Better, faster, stronger. *Bioinformatics* 36, 5424–5431. doi:10.1093/bioinformatics/btaa1029

Privé, F., Aschard, H., Ziyatdinov, A., and Blum, M. G. B. (2018). Efficient analysis of large-scale genome-wide data with two R packages: Bigstatsr and bigsnpr. *Bioinformatics* 34, 2781–2787. doi:10.1093/bioinformatics/bty185

Purcell, S., and Chang, C. (2015). Plink 2.0. Available at: www.cog-genomics.org/plink/2.0/.

Qian, J., Tanigawa, Y., Du, W., Aguirre, M., Chang, C., Tibshirani, R., et al. (2020). A fast and scalable framework for large-scale and ultrahigh-dimensional sparse regression with application to the UK Biobank. *PLOS Genet.* 16, e1009141. doi:10.1371/journal.pgen.1009141

R Core Team (2021). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Sabatine, M. S. (2019). PCSK9 inhibitors: Clinical evidence and implementation. *Nat. Rev. Cardiol.* 16, 155–165. doi:10.1038/s41569-018-0107-8

Sanna, S., Li, B., Mulas, A., Sidore, C., Kang, H. M., Jackson, A. U., et al. (2011). Fine mapping of five loci associated with low-density lipoprotein cholesterol detects variants that double the explained heritability. *PLoS Genet.* 7, e1002198. doi:10.1371/journal.pgen.1002198

Sinnott-Armstrong, N., Tanigawa, Y., Amar, D., Mars, N., Benner, C., Aguirre, M., et al. (2021). Genetics of 35 blood and urine biomarkers in the UK biobank. *Nat. Genet.* 53, 185–194. doi:10.1038/s41588-020-00757-z

Staerk, C., Kateri, M., and Ntzoufras, I. (2021). High-dimensional variable selection via low-dimensional adaptive learning. *Electron. J. Statistics* 15, 1797. doi:10.1214/21-ejs1797

Staerk, C., and Mayr, A. (2021). Randomized boosting with multivariable base-learners for high-dimensional variable selection and prediction. *BMC Bioinforma.* 22, 441. doi:10.1186/s12859-021-04340-z

Strömer, A., Staerk, C., Klein, N., Weinhold, L., Titze, S., and Mayr, A. (2022). Deselection of base-learners for statistical boosting—With an application to distributional regression. *Stat. Methods Med. Res.* 31, 207–224. doi:10.1177/09622802211051088

Tanigawa, Y., Qian, J., Venkataraman, G., Justesen, J. M., Li, R., Tibshirani, R., et al. (2022). Significant sparse polygenic risk scores across 813 traits in UK biobank. *PLOS Genet.* 18, e1010105–e1010121. doi:10.1371/journal.pgen.1010105

Thomas, J., Hepp, T., Mayr, A., and Bischl, B. (2017). Probing for sparse and fast variable selection with model-based boosting. *Comput. Math. Methods Med.* 2017, 1421409–1421418. doi:10.1155/2017/1421409

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Methodol.* 58, 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x

Trynka, G., Wijmenga, C., and van Heel, D. A. (2010). A genetic perspective on coeliac disease. *Trends Mol. Med.* 16, 537–550. doi:10.1016/j.molmed.2010.09.003

van de Geer, S. A. (2008). High-dimensional generalized linear models and the lasso. *Ann. Statistics* 36, 614–645. doi:10.1214/009053607000000929

Vilhjálmsson, B., Yang, J., Finucane, H., Gusev, A., Lindström, S., Ripke, S., et al. (2015). Modeling linkage disequilibrium increases accuracy of polygenic risk scores. *Am. J. Hum. Genet.* 97, 576–592. doi:10.1016/j.ajhg.2015.09.001

Wei, Z., and Li, H. (2007). Nonparametric pathway-based regression models for analysis of genomic data. *Biostatistics* 8, 265–284. doi:10.1093/biostatistics/kxl007

Wood, A. R., Esko, T., Yang, J., Vedantam, S., Pers, T. H., Gustafsson, S., et al. (2014). Defining the role of common variation in the genomic and biological architecture of adult human height. *Nat. Genet.* 46, 1173–1186. doi:10.1038/ng.3097

Yang, I. V., Lozupone, C. A., and Schwartz, D. A. (2017). The environment, epigenome, and asthma. *J. Allergy Clin. Immunol.* 140, 14–23. doi:10.1016/j.jaci.2017.05.011

Yang, J., Bakshi, A., Zhu, Z., Hemani, G., Vinkhuyzen, A. A. E., Lee, S. H., et al. (2015). Genetic variance estimation with imputed variants finds negligible missing heritability for human height and body mass index. *Nat. Genet.* 47, 1114–1120. doi:10.1038/ng.3390

Yang, J., Benyamin, B., McEvoy, B. P., Gordon, S., Henders, A. K., Nyholt, D. R., et al. (2010). Common snps explain a large proportion of the heritability for human height. *Nat. Genet.* 42, 565–569. doi:10.1038/ng.608

Zhang, Q., Privé, F., Vilhjálmsson, B., and Speed, D. (2021). Improved genetic prediction of complex traits from individual-level data or summary statistics. *Nat. Commun.* 12, 4192. doi:10.1038/s41467-021-24485-y

Zou, H., and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 67, 301–320. doi:10.1111/j.1467-9868.2005.00503.x