Briefings in Bioinformatics, 2022, **23(2)**, 1–19

https://doi.org/10.1093/bib/bbac039

Review

PGS-server: accuracy, robustness and transferability of polygenic score methods for biobank scale studies

Sheng Yang and Xiang Zhou

Corresponding authors. Sheng Yang, Department of Biostatistics, School of Public Health, Nanjing Medical University, Nanjing, Jiangsu 211166, China. E-mail: yangsheng@njmu.edu.cn; Xiang Zhou, Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA. E-mail: xzhousph@umich.edu

Abstract

Polygenic scores (PGS) are important tools for carrying out genetic prediction of common diseases and disease related complex traits, facilitating the development of precision medicine. Unfortunately, despite the critical importance of PGS and the vast number of PGS methods recently developed, few comprehensive comparison studies have been performed to evaluate the effectiveness of PGS methods. To fill this critical knowledge gap, we performed a comprehensive comparison study on 12 different PGS methods through internal evaluations on 25 quantitative and 25 binary traits within the UK Biobank with sample sizes ranging from 147 408 to 336 573, and through external evaluations via 25 cross-study and 112 cross-ancestry analyses on summary statistics from multiple genome-wide association studies with sample sizes ranging from 1415 to 329 345. We evaluate the prediction accuracy, computational scalability, as well as robustness and transferability of different PGS methods across datasets and/or genetic ancestries, providing important guidelines for practitioners in choosing PGS methods. Besides method comparison, we present a simple aggregation strategy that combines multiple PGS from different methods to take advantage of their distinct benefits to achieve stable and superior prediction performance. To facilitate future applications of PGS, we also develop a PGS webserver (http://www.pgs-server.com/) that allows users to upload summary statistics and choose different PGS methods to fit the data directly. We hope that our results, method and webserver will facilitate the routine application of PGS across different research areas.

Keywords: polygenic score, genome-wide association study, genetic prediction, complex trait, common disease, cross-ancestry

Introduction

Accurate genetic prediction of complex traits may facilitate disease screening at population scale, improve intervention at an early stage and aid in the development of precision medicine [1–3]. Because most complex traits have a polygenic architecture and are each influenced by thousands of single-nucleotide polymorphisms (SNPs), accurate genetic prediction of complex traits requires modeling genome-wide SNPs and constructing polygenic scores (PGS; [4]). PGS for a trait, in its simplest form, is a weighted summation of genotypes across SNPs with the weights being their estimated genetic effect sizes [5, 6]. PGS is commonly referred to as the polygenic risk score (PRS) or genetic risk score (GRS) when this trait of interest is a binary trait of disease status [7]. PGS are becoming widely applied in the research setting for disease stratification and are becoming adapted towards precision clinical decision across a number of common diseases and disease related complex traits [8–18]. The application of PGS is greatly facilitated by the increasing availability of data from large-scale biobank studies [19]. Commonly available biobank data include UK Biobank (UKB; [20]), Biobank of Japan (BBJ; [21]), China Kadoorie Biobank (CKB; [22]), FINNGEN [23] and All of Us [24], to name a few. Biobank studies often collect a large number of samples, which is crucial for building accurate prediction models. Some biobank studies also focus on samples from a non-European ancestry, which, when paired with biobank studies of predominantly European ancestry, can facilitate the examination of the transferability and portability of PGS across genetic ancestries [25–29].

Because of the importance of PGS and the abundant availability of biobank scale data, many PGS methods have been recently developed to make use of the GWAS summary statistics that are readily available from biobank studies for PGS construction [30]. These PGS methods often rely on a multiple regression modeling framework and make distinct modeling assumptions, either polygenic or sparse, on the SNP effect size distribution underlying the trait of interest [3, 27, 31, 32]. For example, the infinitesimal model, also known as the linear mixed model (LMM) or the best linear unbiased predictor (BLUP), assumes that all SNPs have non-zero effects and that their effect sizes follow a normal

Sheng Yang is an assistant professor at the Department of Biostatistics, Nanjing Medical University, P.R. China. His research interests include statistical genetics and genomics, population genetics and big data.

Xiang Zhou is an associate professor at the Department of Biostatistics, University of Michigan, USA. He is also a member in Center for Statistical Genetics and Center for Precision Health Data Science, University of Michigan. His research focuses on developing statistical methods and computational tools for genetic and genomic studies.

Received: October 2, 2021. **Revised:** December 29, 2021. **Accepted:** January 26, 2022

distribution with a common variance that is shared across SNPs. The infinitesimal model is implemented in multiple software packages that include LDpred2 [33] and summary best linear unbiased prediction (SBLUP; [34]). Similarly, lassosum [35], determination Bayesian sparse linear mixed model (DBSLMM; [27]), latent Dirichlet process regression (DPR; [36]) and polygenic risk score continuous shrinkage (PRS-CS; [37]), assume that all SNPs have non-zero effects and that each SNP effect size follows a normal distribution but with a SNP-specific variance. A SNP-specific variance leads to a scale-normal mixture distribution on the SNP effect sizes, which induces adaptive shrinkage on the SNP effect estimates, resulting in proper shrinkage of small effect estimates without over-shrinkage of the large effect estimates. In contrast to these polygenic models, sparse models assume that only a small proportion of SNPs have non-zero effects. For example, the Bayesian variable selection regression (BVSR) model, implemented as LDpred2-sp [33], specifies a point-normal distribution on the SNP effect sizes. Similarly, SBayesR [38] relies on a mixture of three normal distributions along with a point mass at zero to induce sparsity on the SNP effect estimates. Besides the above model based PGS methods, multiple PGS methods that were initially described as an algorithm can also be viewed as making implicit modeling assumptions on the SNP effect sizes. For example, the most commonly used PGS method, clumping and threshold (CT) [5], relies on linkage disequilibrium (LD) clumping and *P*-value threshold to select a subset of approximately independent SNPs with strong association evidence for PGS construction. The CT strategy ensures a sparse set of SNPs to be used for constructing PGS and thus corresponds to making a sparse assumption on the SNP effect sizes. Similarly, stacked CT (SCT; [39]) extends CT by incorporating a penalized regression to examine an extended set of hyper-parameters for more effective SNP selection.

Given the large number of recently developed PGS methods, one naturally wonders which PGS method one should choose for any given trait of interest. Answering this question remains difficult because only a limited number of comparison studies have been carried out to evaluate the performance of PGS methods. Existing comparative studies are often restricted to a small number of PGS methods and a small number of examined traits and are often carried out within the same genetic ancestry that cannot be used to examine the robustness and transferability of PGS methods across ancestries [40–42]. Here, we perform a comprehensive comparative study on 12 commonly used PGS methods for 50 phenotypes that include 25 quantitative traits and 25 binary traits. We evaluate the performance of PGS methods through internal validations in the UKB as well as cross-study and cross-ancestry validations with external data sources [43]. In addition to method comparison, we present a simple aggregation strategy that combines multiple PGS from different methods to take advantage of their distinct benefits to achieve stable and superior prediction performance. To facilitate future applications of PGS, we also develop a PGS webserver that allows users to upload their own GWAS summary statistics and choose different PGS methods to fit the data directly on the server. Together, we hope that our results can serve as an important guideline for practitioners to choose PGS methods and that our webserver can serve as an important tool for analysts to carry out routine PGS applications.

Methods

Compared PGS methods

We compared a total 12 different PGS methods in the present study. The compared methods include CT, DBSLMM, lassosum, LDpred2-auto, LDpred2-inf, LDpred2-nosp, LDpred2-sp, non-parametric shrinkage (NPS), PRS-CS, SbayesR, SBLUP and SCT. All 12 methods use GWAS summary statistics as input (Table 1). We describe the fitting of these methods in detail below.

CT [5, 44] relies on informed clumping and *P*-value thresholding to select SNPs for PGS construction. We used the *bigsnpr* R package (v.1.4.4) to perform clumping and thresholding for CT. Clumping and thresholding in CT are determined by three hyper-parameters: the *P*-value threshold for selecting the significant SNPs, and the window size and r^2 for LD based SNP clumping. Following [39], we explored different combinations of the three hyper-parameters and selected the optimal parameter combination for each trait through cross-validation. Specifically, we considered 50 different choices for the *P*-value threshold. These *P*-value threshold choices are evenly spaced on the logarithmic scale between the minimum and maximum marginal *P*-values for the trait of interest obtained in the training data. We considered four different choices for the window size (50, 100, 200 and 500 Kb) and seven different choices for r^2 (0.01, 0.05, 0.1, 0.2, 0.5, 0.8 and 0.9). We fitted CT model for each hyper-parameter combination in the training data and selected the optimal combination based on Pearson correlation (R^2; for quantitative traits) or area under the curve (AUC; for binary traits) in the validation data. With the selected optimal hyper-parameter combination, we evaluated the performance of the resulting PGS in the test data.

DBSLMM [27] uses all SNPs for PGS construction. It categorizes SNPs based on their effect sizes into two groups: a group of large effect SNPs and a group of small effect SNPs. DBSLMM effectively places different effect size shrinkages on the two groups of SNPs separately to achieve adaptive shrinkage. Following [27], we used the clumping and thresholding procedure in PLINK (v.1.90b6.9) to select the large effect SNPs setting r^2 to be 0.2. Afterwards, we used the DBSLMM software (v.0.3) to fit the DBSLMM model. DBSLMM contains two hyper-parameters, which is the SNP heritability explained by small effect SNPs and the *P* threshold. We considered three choices for the *P*-value threshold

Table 1. List of compared PGS methods

No.	Methods	Categorization	Parameter tuning	Implementation language	Year of publication	References
1	CT	Non-model-based	Yes	R/Rcpp	2009, 2019	[5, 39]
2	DBSLMM	Polygenic	Yes	R/cpp	2020	[27]
3	lassosum	Polygenic	Yes	R/Rcpp	2017	[35]
4	LDpred2-auto	Polygenic	No	R/Rcpp	2020	[33]
5	LDpred2-inf	Polygenic	No	R/Rcpp	2020	[33]
6	LDpred2-nosp	Polygenic	Yes	R/Rcpp	2020	[33]
7	LDpred2-sp	Sparse	Yes	R/Rcpp	2020	[33]
8	NPS	Non-model-based	Yes	R + cpp	2020	[45]
9	PRS-CS	Polygenic	Yes	Python	2019	[37]
10	SBLUP	Polygenic	No	Cpp	2017	[34]
11	SbayesR	Sparse	No	Cpp	2019	[38]
12	SCT	Non-model-based	Yes	R/Rcpp	2019	[39]

The table lists standard properties of each PGS method that include its categorization (3rd column), parameter tuning (4th column), implementation language (5th column), its year of publication (6th column) and the corresponding reference citations (7th column). CT: clumping and threshold; DBSLMM: tuning version of determination Bayesian sparse linear mixed model (DBSLMM); LDpred2-auto: automatic version of LDpred2; LDpred2-inf: infinitesimal version of LDpred2; LDpred2-nosp: non-sparse version of LDpred2; LDpred2-sp: sparse version of LDpred2; NPS: non-parametric shrinkage; SBLUP: summary best linear unbiased prediction; SbayesR: summary BayesR; SCT: stacked clumping and threshold.

(1e−4, 1e−5 and 1e−6). Following [27], we also considered three SNP heritability choices for DBSLMM in the form of {0.7,1, 1.4} × h^2_{LDSC}, where h^2_{LDSC} is the estimated SNP heritability by LD score regression (LDSC) (v.1.0.1) for the trait of interest. Afterwards, we fitted DBSLMM for each choice of the hyper-parameters in the validation data, selected the optimal hyper-parameter based on the validation data and evaluated the performance of the resulting PGS in the test data. Besides the tuning version of DBSLMM, we also examined the original automatic version of DBSLMM (DBSLMM$_{\text{auto}}$), which does not require a validation dataset and takes the SNP heritability and P threshold choice of h^2_{LDSC} and 1e−6, respectively. We fitted DBSLMM-auto in the training data and evaluated the performance of the resulting PGS in the test data.

Lassosum [35] uses lasso to select SNPs and construct PGS. We used the *lassosum* R package (v.0.4.5) for model fitting. Lassosum contains two hyper-parameters: the penalty parameter (λ) in the lasso regression and the shrinkage parameter (s) used for computing the SNP correlation matrix in the reference panel. Following [35], we considered four choices of s (0.2, 0.5, 0.9 and 1) and 20 choices of λ that are evenly spaced on the logarithmic scale between log(0.001) and log(0.1). We fitted lassosum for each hyper-parameter combination in the reference panel, selected the optimal combination in the validation data and evaluated the performance of the resulting PGS in the test data.

LDpred2 [33] is an updated version of LDpred that can be used to construct PGS using different models. LDpred2 is implemented in the *bigsnpr* R package (v.1.4.4), which we used for model fitting. Following [33], we examined four different models implemented in LDpred2 described as follows. (i) LDpred2-inf is the infinitesimal model that is fitted based on an analytic solution. (ii) LDpred2-nosp fits a BVSR model and selects a small proportion of SNPs to construct PGS. LDpred2-nosp contains two hyper-parameters that include the proportion of causal variants P and the SNP heritability h^2. LDpred2-nosp explores different combinations of the two hyper-parameters on a pre-selected set of grid values and determines the optimal hyper-parameter combination through cross-validation. (iii) LDpred2-sp fits the same model as LDpred2-nosp but sets the effect of some SNP to be zero in the estimation process and thus becomes sparse. (iv) LDpred2-auto fits the same model as LDpred2-nosp but automatically estimates P and h^2 from the reference panel. Due to memory and computational constraints, for each LDpred2 method, we followed [33] to fit one chromosome at a time and combine the resulting PGS across chromosomes as the final PGS for the method. For LDpred2-sp and LDpred2-nosp, we considered different hyper-parameter choices following the software recommendation. Specifically, for P, we considered 21 different values that range from 10^{-5} to 1 on the log-scale. For h^2, we considered three different values in the form of {0.7,1, 1.4} × h^2_{LDSC}. We fitted all four methods in the training data, selected the optimal hyper-parameter combination for LDpred2-sp and LDpred2-nosp in the validation data and evaluated the performance of the resulting PGS in the test data.

NPS [45] transforms SNPs into an orthogonal eigenlocus space, groups eigenlocus into different partitions based on their marginal effect sizes and shrinks the eigenlocus effect sizes in different partitions differently. We used *NPS* R software (v.0.1) to fit NPS and used its default settings with the window size set to be 4 kb following [45]. We fitted NPS model in the training data and tested PGS performance in the test data.

PRS-CS [37] places a CS prior on SNP effect sizes. We used the PRS-CS python package for model fitting. Following [37], we set the hyper-parameter a in PRS-CS to the default value of 1, set the hyper-parameter b to the default value of 0.5 and inferred the global scaling hyper-parameter ϕ among a set of four choices {10^{-6}, 10^{-4} and 0.01,1}. We also examined the automatic version of PRS-CS, referring to PRS-CS$_{\text{auto}}$, which inferred ϕ based on the reference panel alone. We fitted PRS-CS in the training data and evaluated the performance of the resulting PGS in the test data.

SbayesR [38] is a summary statistics version of BayesR. It selects SNPs with non-zero effects using a sparsity inducing prior that consists of a point mass at zero along with a mixture of normal distributions. We used the GCTB software (v.2.02) to fit SbayesR. Following [38], we set the weights of the four normal components ('–*pi*') to the default values of $\{0.95, 0.02, 0.02 \text{ and } 0.01\}$ and set the four scaling variance parameters ('–*gamma*') to the default values of $\{0,\ 0.01,\ 0.1 \text{ and } 1\}$. We constructed the SNP LD matrix using the '–*make-shrunk-ldm*' option, again with the default settings (effective population size = 11 400; genetic map sample size = 183 and shrinkage hard threshold = 10^{-5}). We set the MCMC chain length to be 10 000 with an additional 2000 burn-in iterations. We fitted SbayesR model in the training data and evaluated the performance of the resulting PGS in the test data.

SBLUP [34] fits the same infinitesimal model as LDpred2-inf but with a slightly different fitting algorithm. We used the GCTA software (v.1.93.2) to fit SBLUP and used h^2_{LDSC} as the SNP heritability input. SBLUP also requires users to specify a LD window size that is used to construct the SNP correlation matrix in the reference panel. We fitted SBLUP model in training data and evaluated the performance of the resulting PGS in the test data.

SCT [39] extends CT by searching over a larger parameter space set up by four hyper-parameters. We used *bigsnpr* R package (v.1.4.4) to fit SCT with default settings. We fitted SCT for all combinations of hyper-parameters in the training data, applied an elastic net based penalized regression on the resulting PGS from these hyper-parameter combinations in the validation data to calculate the final SNP weights and evaluated the performance of the resulting PGS in the test data.

Besides the 12 PGS methods, we also explored a novel strategy of aggregating different PGS constructed from all 12 methods into a single PGS for phenotype prediction. The aggregated PGS, which is referred to as the PGSagg, is a weighted summation of the 12 PGS, with the weights determined in a separate/second validation data. In particular, we randomly selected 5% of the entire sample to create a second validation data. We then fitted each of the 12 PGS methods in the training data, tuned their hyper-parameters in the validation data if needed, estimated the weights for the 12 PGS in the second validation data by fitting a standard linear regression and evaluated PGS performance in the remaining samples of the test data.

Note that the majority of the compared PGS methods, with the only exception of CT and SCT, explicitly model the SNP correlation structure induced by LD during PGS construction. These methods obtained the SNP correlation matrix from a reference panel using three different approaches. Due to the constraint of memory and time, we set different LD window size for LDpred2 and SBLUP, including Hapmap phase 3 (HM3) version and the imputed UK BiLEVE version of the genotype data (details of the genotype data are provided in the next section). In particular, DBSLMM, lassosum and PRS-CS constructed a block-diagonal SNP correlation matrix with varying block sizes following [46]; SBLUP constructed a block-diagonal SNP correlation matrix with a fixed block size that equals to 1 Mb for non-dense SNP set and 0.2 Mb for dense SNP set; LDpred2 set different LD window size to be 3 cM for non-dense SNP set and 0.2 Mb for dense SNP set; SBLUP constructed a block-diagonal SNP correlation matrix with a fixed block size that equals to 1 Mb for non-dense SNP set and 0.2 Mb for dense SNP set; NPS constructed a SNP correlation matrix with overlapped blocks where each block consisted of 4000 SNPs and the shift window size was set to 0, 1000, 2000 and 3000 SNPs and SbayesR constructed a banded SNP correlation matrix using the shrinkage procedure presented in [47].

In the analysis, we also recorded the computation time and memory usage for the compared PGS methods. For SCT, because it only includes one additional step on top of CT and the additional step does not incur much computational cost, we simply used the computing cost of CT for SCT. For LDpred2-sp and LDpred2-nosp, because they are fitted using the same function and are not separatable from each other, we recorded their combined computing cost and denoted it as LDpred2. In total, we recorded computation time and memory usage for eight PGS methods.

UKB internal cross-validation

We evaluated the performance of PGS methods for 50 traits in the UKB data through Monte Carlo cross-validation. The 50 traits consist of 25 quantitative traits and 25 binary traits. The 25 quantitative traits include standing height (SH), bone mineral density (BMD), high density lipoprotein (HDL), basal metabolic rate (BMR), platelet count (PLT), body mass index (BMI), body fat percentage (BFP), age at menarche (AM), hip circumference (HC), red blood cell count (RBC), trunk fat percentage (TFP), RBC distribution width (RDW), waist circumference (WC), eosinophils count (EOS), total triglycerides (TG), white blood cell count (WBC), forced vital capacity (FVC), forced expiratory volume (FEV) versus FVC ratio (FFR), FEV, waist–hip ratio (WHR), systolic blood pressure (SBP), total cholesterol (TC), low density lipoprotein (LDL), birth weight (BW) and sodium in urine (SU). The 25 binary traits include type I balding (T1B), qualification (QU), hypertension (HT), tanning ability (TA), smoking status (SS), myxedema (MY), ever smoked (ES), salt added to food (SAF), morning person (MP), asthma (AS), dried fruit intake (DFI), snoring (SN), tense (TE), angina (AN), headache (HA), coronary artery disease (CAD), prostate cancer (PRCA), gout (GO), fresh fruit intake (FFI), type II diabetes (T2D), supplementary vitamin and mineral (VMS), depression (MDD), breast cancer (BRCA), rheumatoid arthritis (RA) and osteoarthritis (OA). Among the 25 binary traits, for the 11 diseases, we treated either self-reported or ICD10 cases as 1 and others as 0 following [39]. For TA, we treated 'get very tanned' as 1 and the others as 0. For MP, we treated 'definitely a morning person' as 1 and the others as 0. Details of the phenotypes analyzed in the paper are shown in Supplementary Table S1 for quantitative traits and Supplementary Table S2 for binary traits.

We performed quality control (QC) following the steps described in [27]. Specifically, for sample QC, we retained individuals (i) who have genotypes successfully measured, (ii) who are included in the genotype principal component (PC) computation and (iii) who have a white British ancestry. In addition, we excluded individuals (i) who have more than 10 putative third-degree relatives based on the kinship table, (ii) who have sex chromosome aneuploidy and (iii) who are redacted and thus do not have a corresponding ID in the phenotype data. For SNP QC, we retained SNPs with a high genotype calling confidence, as is evident by the maximum probability across the three genotypes being larger than 0.9. We filtered out SNPs (i) with a minor allele frequency (MAF) < 0.01, (ii) with a Hardy–Weinberg equilibrium (HWE) test P-value $<10^{-7}$, (iii) with an imputation information score < 0.8, (iv) with a proportion of missingness $(P_m) > 0.05$ or (v) that are a duplicated SNP. We retained a total of 337 129 EUR individuals and analyzed two SNP sets: the HM3 SNP set with 1 246 534 SNPs that are measured in the HM3 and the BiLEVE SNP set with 9 116 018 SNPs that are imputed based on the BiLEVE study. For the HM3 SNP set, we also examined a variation of this SNP set by including 503 836 additional low MAF SNPs (MAF between 0.005 and 0.01) to examine the benefits of including low MAF SNPs for PGS accuracy. We only fitted eight PGS methods (CT, DBSLMM, lassosum, LDpred2-auto, LDpred2-inf, LDpred2-nosp, LDpred2-sp and SCT) on the SNP set with low MAF SNPs due to computational constraint.

For each phenotype in turn, we performed Monte Carlo cross-validation to evaluate the performance of different PGS methods [27, 38]. Specifically, we first randomly selected 500 individuals (250 males and 250 females) who have measurements for all traits to serve as the reference panel. The reference panel is used to estimate the SNP correlation matrix. We then split the remaining data into four sets: a training data, a validation data, a second validation data and a test data. We performed data split in two different ways. In the first way, we randomly selected 1000 samples to serve as the first validation data and another 1000 samples to serve as the second validation data. We then split the remaining samples into a training data with 80% samples and a test data with 20% samples. The relatively small validation sample size allows us to carry out the comparison for different PGS methods efficiently with our limited computational resource. Indeed, many PGS methods (e.g. CT, DBSLMM, lassosum, LDpred2, NPS and SCT) would require >64GB memory if the validation data comes with a larger sample size, especially in the dense SNP setting. In the second way, we randomly selected 80% samples into the training data, 5% samples into the first validation data, 5% samples into the second validation data and the remaining 10% samples into the test data. In particular, the number of samples for the quantitative traits now ranges from 8826 (for trait AM) to 16 987 (for trait BMI), which is 8.83–16.99 times higher than that in the validation data from the first way of data split. The number of samples for the binary traits now ranges from 7740 (for trait PRCA) to 16 985 (for trait SAF). The second way of data split ensures a relatively large sample size in the validation dataset to ensure more potentially accurate PGS performance. We presented the main results based on the second way of data split and presented the other way of data split as supplementary results.

With either way of data split, we repeated the process five different times for Monte Carlo cross-validation. In each replication, we fitted each of the 12 PGS methods in the training data, tuned their hyper-parameters in the validation data if needed, estimated the weights for the 12 PGS in the second validation data by fitting a standard linear regression and evaluated PGS performance in the remaining samples of the test data. Specifically, in the training data, we obtained marginal z-scores through linear regression implemented in the GEMMA software [48]. Specifically, for each quantitative trait, we first fitted a linear regression to remove the effects of the top 10 genotype PCs and sex and obtained phenotype residuals. We transformed phenotype residuals to a standard normal distribution through quantile–quantile transformation. We then examined one SNP at a time and obtained its marginal z-score for the trait using a standard linear regression. For each binary trait, in the main analysis, we directly fitted a linear regression model for each SNP in turn by treating the top 10 genotype PCs and sex as covariates to obtain the marginal z-scores. We fitted different prediction methods in the training data using the marginal z-scores. For binary traits, we also examined the accuracy difference in PGS constructed by the models fitted by linear versus that fitted by logistic regression. To do so, we focused on two non-model-based methods, CT and SCT, and fitted them using the marginal SNP effect size estimates obtained from the logistic regression instead of the linear regression, for all 25 binary traits in the internal cross-validation. We did not examine the model-based methods because the model-based methods all explicitly use a multiple linear regression model for PGS construction. In the analysis, we also applied logistic regression to tune the hypermeters in CT and SCT.

After model fitting and tuning, we supplied the estimated SNP effects from different PGS methods to the test data to construct PGS using the *score* function in PLINK [49]. For quantitative traits, we evaluated the performance of different PGS methods in the test data using R^2 and mean standard error (MSE; calculated by *Metrics* R package v.0.1.4). For binary traits, we evaluated the performance of different PGS methods in the test data using AUC (calculated by *pROC* R package v.1.15.3) and

Nagelkerke pseudo R^2 (calculated by *rms* R package v.6.1). After obtaining these metrics, we calculated the relative performance of each PGS method by contrasting it with respective to the best method in each validation. Specifically, for R^2 and Nagelkerke pseudo R^2, we calculated the relative performance of each method as the ratio between the performance of each method and that of the best method in the replicate:

$$Relative\ Performance = \frac{Performance_{each\ method}}{Performance_{best\ method}} \tag{1}$$

For AUC, we calculated the relative performance of each method as:

$$Relative\ Performance = \frac{AUC_{each\ method} - 0.5}{AUC_{best\ method} - 0.5} \tag{2}$$

For MSE, we followed [31] and calculated the relative performance of each method as:

$$Relative\ Performance = \frac{MSE_{each\ method} - MSE_{intercept}}{MSE_{best\ method} - MSE_{intercept}} \tag{3}$$

where $MSE_{intercept}$ is the MSE calculated with the simple regression model with only an intercept term. Note that the relative performance calculated based on R^2, AUC or Nagelkerke pseudo R^2 is between 0 and 1, whereas the relative performance calculated based on MSE is a value below 1 where a negative value indicates that the method performances even worse than the intercept model. Following [27, 31], we defined the proportion of the genetic variance explained by large effect SNPs as PGE:

$$\mathrm{PGE} = \frac{Var\left(\mathbf{X}_l\beta_l\right)}{Var\left(\mathbf{X}_l\beta_l + \mathbf{X}_s\beta_s\right)} \tag{4}$$

We donated genotype matrix of large effect SNPs as $\mathbf{X}_l$, effect size of large effect SNPs as β_l, genotype matrix of small effect SNPs as $\mathbf{X}_s$ and effect size of large effect SNPs as β_s.

Besides directly reporting these metrics for evaluating method performance, we also categorized PGS methods into three performance groups for every trait. Specifically, for each trait in turn, we compared the performance of all other PGS methods with the top PGS method using either a paired *t*-test or a Wilcoxon signed-rank test. The paired *t*-test directly tests on the R^2 after Fisher transformation [50] (for quantitative traits), whereas the Wilcoxon signed-rank test tests AUC (for binary traits) from the two compared PGS methods across five test sets, respectively. The PGS methods that have a similar performance as the top PGS method based on a Bonferroni corrected *P*-value threshold of 0.05 (=0.05/12) are categorized into the top performance group. Similarly, we examined the performance of all other PGS methods with the bottom PGS method and categorized PGS methods with similar performance into the bottom performance group. The remaining PGS methods are then categorized into the medium performance group. Note that the top group is defined with respect to each metric. We report the top group categorization in the main results based on R^2 for quantitative traits and AUC for binary traits, though the categorization is quite consistent regardless which metric we use.

External validation on three data categories: AFR, ASA and EUR

Besides evaluating PGS methods through Monte Carlo cross-validations within white British of UKB, we also evaluated their performance in other ancestry groups of UKB and other data sources. We simply refer to these validations as external validations. The external validations were carried out on the 25 quantitative traits, which are more readily available than the binary traits in external data sources. In the external validations, we first fitted different PGS methods in each one of the five training sets of UKB. We then examined their prediction performance in the external datasets that are grouped into three general categories based on ancestry. The three general categories are African (AFR), Asian (ASA) and EUR and are described in detail below.

We obtained 7891 AFR individuals in UKB, defined as Black or Black British, White and Black Caribbean, White and Black African, Caribbean, African or any other Black background. We followed the same procedure described in the previous section for data processing. For all traits, we intersected SNPs from African UKB with the HM3 SNP set from white British UKB and analyzed an overlap set of 760 711 SNPs. We analyzed all 25 quantitative and 25 binary traits for AFR (trait details in Supplementary Tables S3 and S4, see Supplementary Data available online). We evaluated the performance of PGS methods using the individual level data of AFR.

The ASA data consists of two data sources: individuals with Asian ancestry in UKB and BBJ. For UKB, we obtained 3790 individuals with Asian ancestry, defined as Chinese, White and Asian or any other Asian background, and analyzed all 25 quantitative and 25 binary traits (trait details in Supplementary Tables S5 and S6, see Supplementary Data available online). We evaluated the performance of PGS methods using the individual level data of ASA. For BBJ, we obtained summary statistics on 12 quantitative traits that were measured there. The 12 traits include SH_BBJ (*n* = 159 095) [51], HDL_BBJ (*n* = 70 657) [52], PLT_BBJ (*n* = 108 208) [52], BMI_BBJ (*n* = 158 284) [53], AM_BBJ (*n* = 67 029) [54], RBC_BBJ (*n* = 108 794) [52], EOS_BBJ (*n* = 62 076) [52], TG_BBJ (*n* = 105 597) [52], WBC_BBJ (*n* = 107 964) [52], SBP_BBJ (*n* = 136 597) [52], TC_BBJ (*n* = 128 305) [52] and LDL_BBJ (*n* = 72 866) [52] (Supplementary Table S5, see Supplementary Data available online). We evaluated the performance of PGS methods using the summary statistics of BBJ. For all traits, we intersected SNPs from the two data sources with the HM3 SNP set from white British UKB and analyzed an overlap set of 883 764 SNPs.

The EUR data consist of GWAS summary statistics obtained for individuals with European ancestry from two data sources: GWAS-ALTAS and GRASP [55, 56]. In the two data sources, we extracted datasets that contain the analyzed quantitative traits in UKB, that have greater than 100 000 SNPs, that contain summary statistics with allelic information and that are measured only on non-UKB individuals not part of the UKB data. With these criteria, we obtained 25 datasets for 19 quantitative traits that include SH ($n = 253\ 288$) [57], BMD ($n = 66\ 628$) [58], HDL_GLGC ($n = 188\ 577$) [59], HDL_meta ($n = 19\ 840$) [60], HDL_NMR ($n = 24\ 925$) [61], PLT ($n = 4250$) [62], BMI ($n = 253\ 288$) [63], AM_RG1 ($n = 49\ 427$) [64], AM_RG2 ($n = 329\ 345$) [65], HC ($n = 142\ 762$) [4], RBC ($n = 4250$) [62], WC ($n = 142\ 762$) [4], EOS ($n = 4250$) [62], TG_GLGC ($n = 188\ 577$) [59], TG_NMR ($n = 24\ 925$) [61], WBC ($n = 4250$) [62], FVC ($n = 79\ 055$) [66], FFR ($n = 79\ 055$) [66], FEV ($n = 79\ 055$) [66], WHR ($n = 142\ 762$) [4], TC ($n = 188\ 577$) [59], LDL_GLGC ($n = 188\ 577$), LDL_meta ($n = 19\ 840$) [60], LDL_NMR ($n = 24\ 925$) [61] and BW ($n = 69\ 308$) [67]. Among them, SH was adjusted for the top 20 genotype PCs [57]. The four blood measurements (PLT, RBC, EOS and WBC) were adjusted for age, sex and time of blood collection with both a linear and square components [62]. BMD was corrected for age, weight, height, genomic principal components and study-specific covariates [58]. BMI was adjusted for age, age^2 and other covariates such as genotype PCs [63]. The adjusted covariates for AM_RG1 and AM_RG2 in the ReproGen consortium were not described in detail in the original study [64, 65]. The three lung function traits (FVC, FEV and FFR) in the SpiroMeta study were adjusted for age, age^2, sex, height and genotype PCs [66]. The three anthropometric traits (WC, HC and WHR) in the GIANT consortium were adjusted for age, age^2 and study-specific covariates [4]. The four lipid traits (TC, HDL_GLGC, LDL_GLGC and TG_GLGC) in the GLGC consortium were adjusted for age, age^2, sex and genotype PCs [59]. The two lipid traits (HDL_meta and LDL_meta) in the meta-analysis used different adjusted covariates in sub-studies [60]. The three lipid traits (HDL_NMR, LDL_NMR and TG_NMR) were adjusted for age, sex, time from last meal, if applicable, and first 10 principal components from genomic data [61]. BW was adjusted for sex and gestational age [68]. Data details are provided in Supplementary Table S8. For all traits, we intersected SNPs from the 25 datasets with the HM3 SNP set from white British UKB and analyzed an overlap set of 821 361 SNPs.

Because the external data sources contain only summary statistics, we computed R^2 in the test set using summary statistics following the formula in [27] to evaluate PGS performance. Specifically,

$$R = cor\left(\tilde{\mathbf{Y}}, \hat{\tilde{\mathbf{Y}}}\right) = \frac{\frac{1}{\tilde{n}}\tilde{\mathbf{Y}}^T\tilde{\mathbf{X}}\hat{\beta}}{\sqrt{\frac{1}{\tilde{n}}\hat{\beta}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\hat{\beta}}} = \frac{\sqrt{\frac{1}{\tilde{n}}}\tilde{\mathbf{z}}^T\hat{\beta}}{\sqrt{\hat{\beta}^T\boldsymbol{\Sigma}\hat{\beta}}} \qquad (5)$$

where we denoted the unobserved individual-level phenotype vector in the external data as $\tilde{\mathbf{Y}}$, the unobserved individual-level genotype matrix as $\tilde{\mathbf{X}}$, the observed summary statistics in terms of z-scores as $\tilde{\mathbf{z}}$, the estimated SNP weights from the PGS method as $\hat{\beta}$, the sample size as $\tilde{n}$ and $\boldsymbol{\Sigma}$ as the SNP correlation matrix, which is estimated from the reference panel.

Software and PGS web server

Code for fitting different PGS methods and scripts for reproducing all analyses carried out in the present study are available at https://github.com/biostat0903/PGS-Server. A PGS computing web server developed in the present study is available at http://www.pgs-server.com.

Results

PGS method comparison overview

We evaluated the performance of 12 PGS methods on 25 quantitative traits and 25 binary traits through internal and external cross-validations (Figure 1A; details in Methods). An overview of the method comparison workflow is shown in Figure 1. The compared PGS methods can be generally categorized into model-based methods and non-model-based methods. The model-based methods include polygenic methods that assume all SNPs to have non-zero effects and sparse methods that assume a small set of SNPs to have non-zero effects. The polygenic methods can be further categorized into two subgroups based on their detailed modeling assumptions on the effect sizes [69]: a subgroup of methods that assume a simple normality assumption on the effect sizes and effectively perform a global shrinkage on the effects regardless of their sizes; and another subgroup of methods that place a local shrinkage on each SNP to induce a heavy tailed distribution on the effects sizes and shrink the small effects adaptively towards zero more than the large effects (Figure 1B). The non-model-based methods include all the remaining methods that do not make an explicit modeling assumption. Note that we followed [3] to categorize the model-based methods based on their modeling assumptions rather than the underlying fitting algorithms: a method is categorized as a polygenic method per its modeling assumption even though it may yield a sparse solution due to its fitting algorithm; and vice versa.

In the internal validations, we focused on 337 129 white British samples of UKB and performed Monte Carlo cross-validation five times for each of the 50 traits (Supplementary Tables S1 and S2, see Supplementary Data available online). In particular, we trained PGS methods in a training data with 80% individuals, tuned the hyper-parameters in a validation data with 5% individuals, fitted PGSagg in the second validation data with 5% individual and evaluated their performance in a test data with the remaining 10% individuals (Figure 1C). When we fitted PGS models in dense SNP set for 25 quantitative traits, we treated 1000 samples randomly

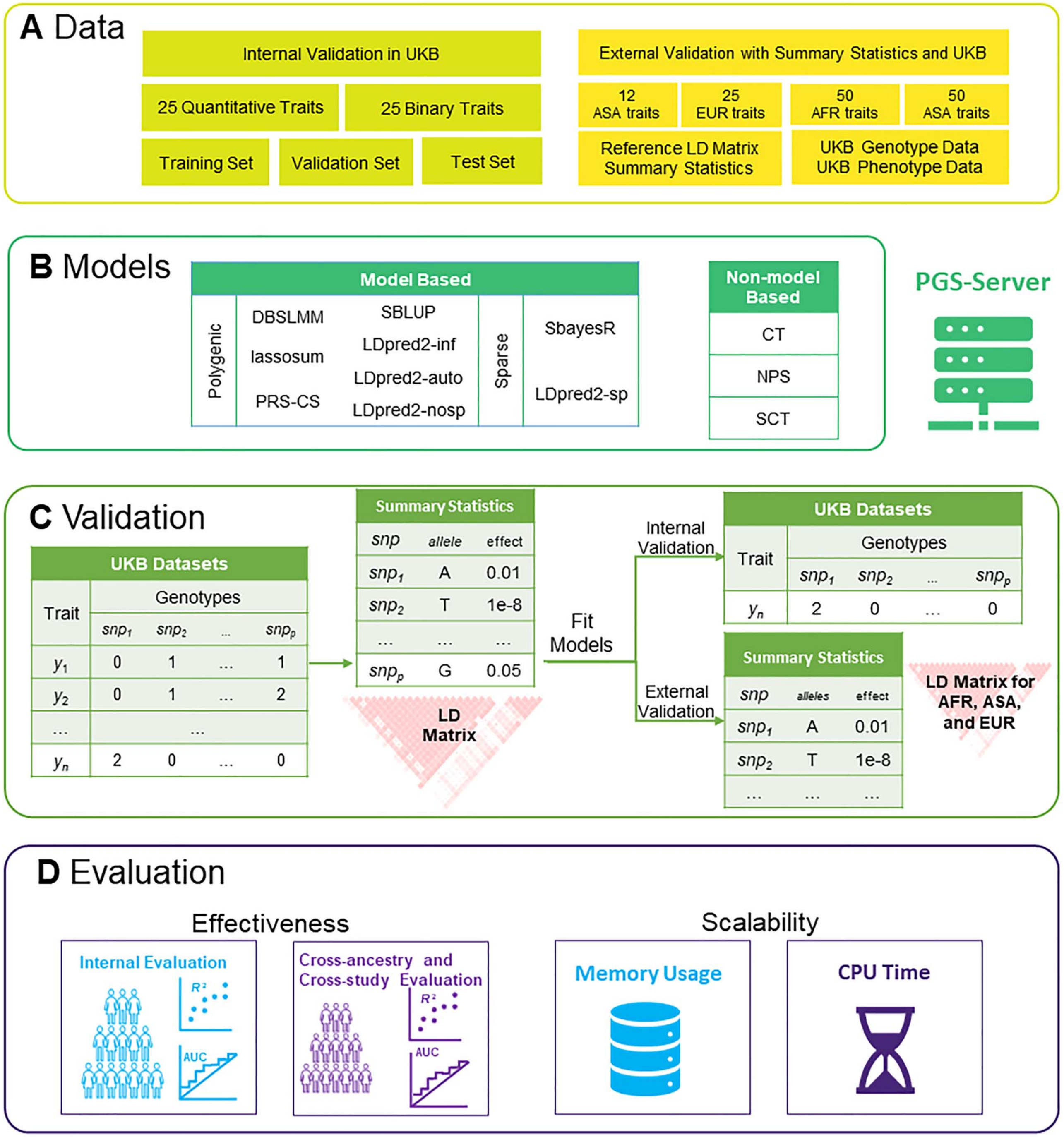

Figure 1. Overview of the comparison workflow in the present study. (**A**) We obtained 50 traits from UKB with European ancestry, 25 from other GWASs with European ancestry and 112 from GWASs with Asian or African ancestries. (**B**) We evaluated the performance of 12 PGS methods that include model-based ones such as polygenic methods and sparse methods as well as non-model-based ones. (**C**) For each examined trait, we divided the UKB data into a training data, a validation data and a test data. We fitted each PGS method in the training data based on GWAS summary statistics and an LD reference matrix, and, if necessary, selected the optimal hyper-parameter combinations in the validation data. We then evaluated the performance of the resulting PGS either in the UKB test data based on individual-level data (internal cross-validations) or external test data based on GWAS summary statistics and individual-level data (external cross-validations). (**D**) We evaluated the performance of PGS methods by Pearson R^2 (for quantitative datasets) or AUC (for binary datasets) in the test data. We also recorded the scalability of each PGS construction method across data splits by recording the physical memory requirement and computation time.

selected from the first validation set as the validation data.

In the validation, we examined the influence of SNP density on prediction accuracy using two SNP sets: a non-dense SNP set with 1 246 534 SNPs characterized in the HM3 SNP set and a dense SNP set with 9 116 018 SNPs measured on the imputed version of UK BiLEVE axiom array (BiLEVE SNP set; [70]). We applied all PGS methods to the HM3 SNP set and applied most PGS methods to the UK BiLEVE SNP set (except for NPS, PRS-CS and SbayesR due to their heavy computational burden). We evaluated the performance of PGS methods in the test data using the following absolute metrics: R^2 or MSE for quantitative traits, and AUC or pseudo/Nagelkerke R^2 for binary traits (Figure 1D). We also calculated the relative performance of each method by contrasting its performance value with respect to that of the best method in each cross-validation for each trait. We present the main results based on the relative performance to facilitate the comparison of method performance across traits, but we also present absolute performance results in the Supplementary Figures.

In the external validations, we focused on quantitative traits and fitted PGS methods in the same UKB white British training data but evaluated their performance in three other data sources (Figure 1C). The three other data sources include AFR, which consists of 7891 African samples in UKB (Supplementary Tables S3 and S4, see Supplementary Data available online); ASA, which consists of two sub-datasets, including one sub-data with 3790 Asian samples in UKB and another sub-data with 62 076–159 095 individuals in BBJ (Supplementary Tables S5 and S6, see Supplementary Data available online) and EUR, which consists of 4250–329 345 individuals of European ancestry from multiple external data sources other than UKB (Supplementary Table S7, see Supplementary Data available online). Because BBJ data sources contain summary statistics, we computed R^2 in the test set using summary statistics following the Equation (5) to evaluate PGS performance [27]. We calculated the relative performance of each method by contrasting its performance value with respect to that of the best method in each cross-validation for each trait (Figure 1D).

Internal cross-validations in UKB

We first examined the performance of PGS methods through internal cross-validation. For each trait in turn, we evaluated the performance of methods in the test data (Supplementary Figures S1B, D, S2 and S3 for quantitative traits; Supplementary Figures S4B, D, S5 and S6 for binary traits, see Supplementary Data available online) and ranked them based on the average performance across Monte Carlo cross-validation (Figure 2A and B for quantitative traits; Figure 2C and D for binary traits). Across traits, we found that two of the polygenic methods that perform local shrinkage on the effect sizes achieved the best performance. Specifically, the relative accuracy of DBSLMM is on average 91.21 and 97.65% for quantitative and binary traits, respectively; and that of lassosum is 89.98 and 86.78% for quantitative and binary traits, respectively. The other local shrinkage method PRS-CS also performs reasonably well: its relative accuracy is on average 68.21 and 91.98% for quantitative and binary traits, respectively. The performance of the local shrinkage polygenic methods is generally followed by the global shrinkage polygenic methods. For example, the relative accuracy of LDpred2-nosp is on average 92.70 and 95.03% for quantitative and binary traits, respectively; and that of LDpred2-inf is 78.74 and 92.15%, respectively. In contrast, the sparse methods that assume a small proportion of SNPs to have non-zero effects often do not perform as well as the polygenic methods. For example, the relative accuracy of SbayesR is on average 57.77 and 89.21% in quantitative and binary traits, respectively. As one might expect, non-model-based methods, such as CT and NPS, often do not perform as well as the model-based methods. Importantly, the ranking for most PGS methods is reasonably consistent regardless of which metric we used to evaluate their performance (Supplementary Figures S2, S5 and S7, see Supplementary Data available online).

Besides directly reporting the ranking of PGS methods based on the evaluation metrics, we categorized PGS methods into three performance groups (top, medium and bottom) for every trait (Supplementary Figure S1A and C for quantitative traits; Supplementary Figure S4A and C for binary traits; categorization details in Methods). Method categorization revealed largely consistent results as compared with direct ranking of methods. For example, both lassosum and DBSLMM perform the best: DBSLMM is in the top group for 19 quantitative traits and 23 binary traits, whereas lassosum is in the top group for 14 quantitative traits and seven binary traits. Their performance is followed by other local shrinkage that include PRS-CS (top for three quantitative traits and 13 binary traits) and global shrinkage methods that include LDpred2-nosp (top for 20 quantitative traits and 18 binary traits), LDpred2-auto (top for one quantitative trait and seven binary traits), LDpred2-inf (top for one quantitative trait and 15 binary traits) and SBLUP (top for two quantitative traits and 15 binary trait). The performances of these polygenic methods are followed by sparse methods such as LDpred2-sp (top for 15 quantitative trait and 21 binary trait) and SbayesR (top for nine binary traits). Some non-model-based methods such as SCT (top for 12 quantitative traits and five binary traits) and CT (top for five quantitative traits) perform reasonably well while NPS is top for no trait.

We carefully examined factors that influence the performance of PGS methods other than their modeling assumptions on the SNP effect size distribution. First, we found that the choice of the SNP sets used for fitting PGS methods does not influence much the performance of the majority of PGS methods (Supplementary Figures S8–S11, see Supplementary Data available online). Indeed, most PGS methods, except for LDpred2 methods, either have similar or slightly reduced performance when using the dense SNP set as compared with the non-dense SNP set. However, the performance of the four LDpred2 methods are substantially reduced when using the dense SNP set (Supplementary Figure S12, see Supplementary Data available online), presumably due to the slow convergence of the LDpred2 algorithm when the number of SNPs is large [27]. Second, we found that the performance of PGS methods for a given trait is reasonably highly correlated with the SNP heritability underlying the quantitative and binary

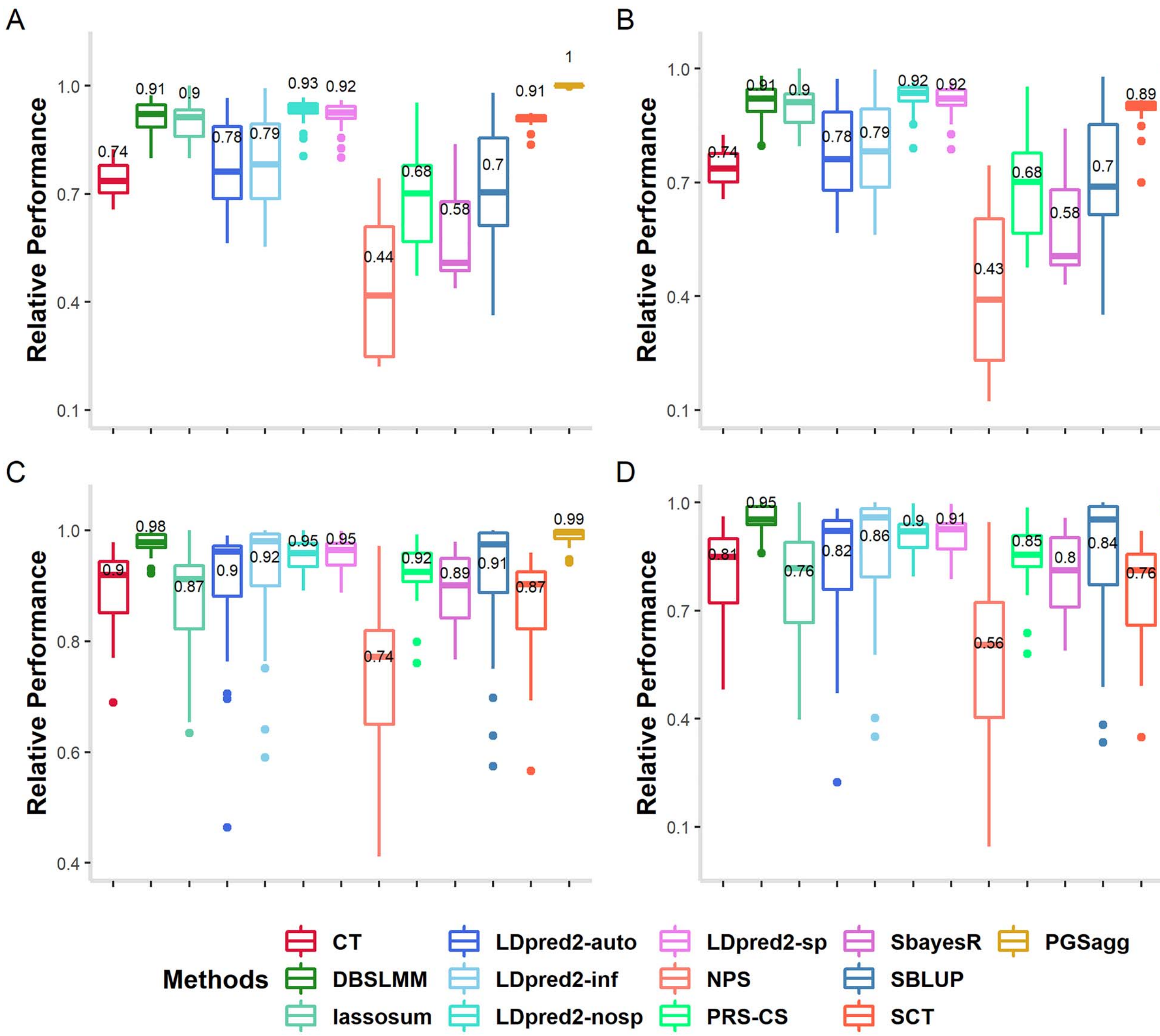

Figure 2. The relative prediction performance of 13 PGS methods for 25 quantitative and 25 binary traits in UKB cross-validations. Compared methods include CT, DBSLMM, lassosum, LDpred2-auto, LDpred2-inf, LDpred2-nosp, LDpred2-sp, NPS, PRS-CS, SbayesR, SBLUP, SCT and PGSagg. (**A**) Boxplot shows the relative performance of each PGS method with respective to the best method in terms of prediction R^2 across validation folds and across 25 quantitative traits. (**B**) Boxplot shows the relative performance of each PGS method with respect to the best method in terms of prediction MSE across validation folds and across quantitative traits. (**C**) Boxplot shows the relative performance of each PGS method with respect to the best method in terms of prediction AUC across validation folds and across 25 binary traits. (**D**) Boxplot shows the relative performance of each PGS method with respect to the best method in terms of the prediction Nagelkerke R^2 across validation folds and across binary traits.

traits (Supplementary Figures S13, S14, S17 and S18, see Supplementary Data available online). Specifically, for quantitative traits, the linear regression coefficients between the prediction R^2 and SNP heritability estimate across traits ranges from 0.336 (for NPS) to 0.613 (for LDpred2-nosp), with a mean value of 0.509 across methods. For binary traits, except for SBLUP, the linear regression coefficients between AUC and SNP heritability estimate across traits ranges from 1.257 (for NPS) to 1.701 (for DBSLMM), with a mean value of 1.562 across methods. In addition, the performance of PGS methods is also reasonably highly correlated with the proportion of the genetic variance explained by large effect SNPs, a term known as PGE (Supplementary Figures S15, S16, S19 and S20, see Supplementary Data available online; [27, 31]). For quantitative traits, the linear regression coefficients between the prediction R^2 and PGE ranges from 0.239 (for NPS) to 0.415 (for DBSLMM), with an average value of 0.329 across methods. For binary traits, the linear regression coefficients between AUC and PGE ranges from 1.146 (for NPS) to 1.539 (for DBSLMM), with an average value of 1.404 across methods. Third, for the same model implemented in the same software, the tuning version that selects hyper-parameter based on a separate validation data often outperforms the automatic version, which infers the hyper-parameters directly in the training data. For example, the default tuning version of DBSLMM is on average 1.13% (median = 1.14%) better than $\text{DBSLMM}_{\text{auto}}$; whereas the default tuning version of PRS-CS is 12.29% (median = 3.48%) better than $\text{PRS-CS}_{\text{auto}}$ (Supplementary Figures S21 and S22, see Supplementary Data available online). Fourth, the prediction performance for the majority of methods remains almost identical regardless of whether low MAF SNPs are included or not: the median AUC improvement by using low MAF SNPs is 0.00% across methods and traits (range: −0.06 to 0.07%; Supplementary Figure S23, see Supplementary Data available online). The only exception is lassosum, whose performance is slightly improved when low MAF SNPs are included to the model, with a median AUC improvement of 7.32% across all traits (range: −0.09 to 53.84%). Finally, for the two non-model-based methods (CT and SCT) that can make use of marginal SNP effect size estimates obtained from the logistic regression and fit PGS using logistic regression, their performances remain almost identical across the 25 binary traits in the internal cross-validation, regardless of whether we used the logistic regression or linear regression (Supplementary Figure S24, see Supplementary Data available online).

Because different PGS methods have different performance for different traits, we explored a simple aggregation strategy, which we refer to as PGSagg, to combine the benefits of all 12 PGS methods (details in Methods). Specifically, we obtained PGS from each of the 12 methods and aggregated them into a single PGS through a weighted summation. The weights in the summation are inferred from a separate validation set (5% of the total sample size). In the analysis, we found that PGSagg works well across all 50 traits and is ranked as the best method for 24 quantitative traits and 12 binary traits. For traits where the aggregation method performs the best, its improvement over the second-best method is on average 4.59%. For traits where the aggregation method does not perform the best, its accuracy loss as compared with the best method is on average 1.92%. The average improvement in prediction accuracy brought by the aggregation strategy over CT, DBSLMM, lassosum, LDpred2-auto, LDpred2-inf, LDpred2-nosp, LDpred2-sp, NPS, SBLUP, SbayesR, PRS-CS and SCT are 23.54, 5.65, 13.78, 22.51, 19.81, 6.12, 6.45, 133.96, 30.26, 45.92, 32.91, 13.02%, respectively. Therefore, the simple aggregation strategy can serve as an effective approach to improve PGS accuracy across traits.

External validations: AFR, ASA and EUR

We evaluated the performance of PGS methods through external validations to examine their robustness and transferability for cross-ancestry prediction and cross-study prediction [71]. Briefly, we fitted different PGS methods in each of the five training/validation datasets in UKB and then examined their performance in each of the three external data sources that include AFR, ASA and EUR. The external data sources include data measured on different genetic ancestry groups (AFR and ASA-UKB) and data collected from separate studies (ASA-BBJ and EUR), allowing us to examine the cross-ancestry and cross-study performance of different PGS methods. The prediction R^2 results from different PGS methods across the five different UKB training data for the three external data are presented in Figure 3 and Supplementary Figures S25–S33. As expected, the performance of all PGS methods decreases in the external validations as compared with the internal validations. The loss of prediction accuracy is particularly apparent in cross-ancestry validation as compared with cross-study validation. Specifically, with the same training data in UKB, PGS methods lose an average of 81.45% accuracy in AFR (median = 81.53%, ranging from 75.42 to 88.12%; Supplementary Figures S27–S28, see Supplementary Data available online), 68.76% in ASA-BBJ (median = 67.88%, ranging from 59.49 to 93.60%; Supplementary Figure S29, see Supplementary Data available online) and 66.61% in ASA-UKB (median = 66.70%, ranging from 56.92 to 77.50%; Supplementary Figures S29 and S30, see Supplementary Data available online), as compared with that in the UKB test set. In contrast, PGS methods lose an average of 44.10% accuracy in the cross-study validation of EUR (median = 40.06%, ranging from 33.50 to 70.07% Supplementary Figure S31, see Supplementary Data available online), as compared with that in the UKB test set.

Importantly, the rank of different PGS methods remains largely similarly to that observed in the internal validations. In particular, we calculated the fraction of traits on which the performance rank of a given PGS method in the external validations changed no greater than two as compared with internal validations. Such fraction ranges from 35.45% (for PRS-CS) to 84.44% (for DBSLMM) with an average fraction of 63.26% (medium = 63.88%), suggesting that the relative performance of most methods stays similar between the internal and external validations. In addition, consistent with internal validations, we found that certain polygenic methods with local shrinkage tend to perform the best. For example, the relative accuracy of three local shrinkage methods (i.e. DBSLMM lassosum and PRS-CS) is on average 84.44, 75.98 and 35.45%, respectively. The relative accuracy of LDpred2-auto and LDpred2-inf is on average 63.33 and 63.88%, respectively. Some non-model-based methods CT and NPS not fare well. The results based on categorizing methods into three performance categories reach similar conclusions (Supplementary Figures S25A, C, E, G, S26A and C, see Supplementary Data available online). Finally, the aggregation approach PGSagg does not work as well in the external validations as compared with the internal validations (Supplementary Figures S32–S33, see Supplementary Data available online). Specifically, PGSagg is ranked as the third among the PGS methods and its relative performance is 70.97%. In addition, PGSagg works better in the binary traits than the quantitative traits. Specifically, the relative performance of PGSagg across the 25 binary traits is on average 84.53% in the AFR ancestry and 92.68% in the ASA ancestry. Although the relative performance of PGSagg across the

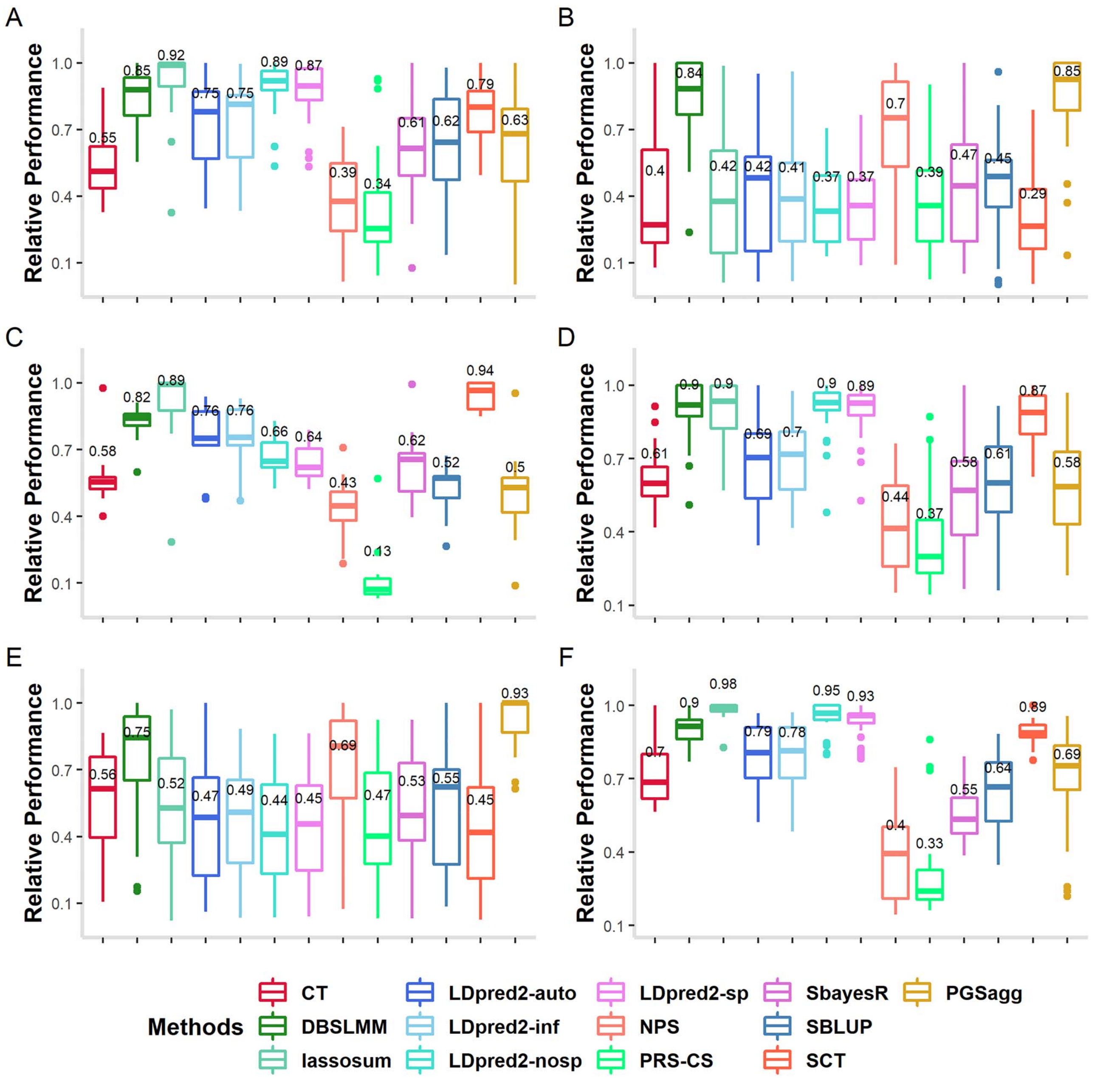

Figure 3. The relative prediction performance of 13 PGS methods for quantitative traits in the external validations. Compared methods include CT, DBSLMM, lassosum, LDpred2-auto, LDpred2-inf, LDpred2-nosp, LDpred2-sp, NPS, PRS-CS, SbayesR, SBLUP, SCT and PGSagg. (**A**) Boxplot shows the relative performance of each PGS method with respective to the best method in terms of prediction R^2 across validation folds and across 25 quantitative traits in AFR ancestry from UKB. (**B**) Boxplot shows the relative performance of each PGS method with respective to the best method in terms of prediction AUC across validation folds and across 25 binary traits in AFR ancestry from UKB. (**C**) Boxplot shows the relative performance of each PGS method with respective to the best method in terms of prediction R^2 across validation folds and across 12 quantitative traits in ASA ancestry from BBJ. (**D**) Boxplot shows the relative performance of each PGS method with respective to the best method in terms of prediction R^2 across validation folds and across 25 quantitative traits in ASA ancestry from UKB. (**E**) Boxplot shows the relative performance of each PGS method with respective to the best method in terms of prediction AUC across validation folds and across 25 binary traits in ASA ancestry from UKB. (**F**) Boxplot shows the relative performance of each PGS method with respective to the best method in terms of prediction R^2 across validation folds and across 25 quantitative traits in EUR ancestry from external summary statistics.

quantitative traits is on average 62.60 and 52.95% in the AFR and the ASA ancestry, respectively.

Computation consumption

We recorded computing time and memory usage for different PGS methods for two example traits that include SH and MDD (Figure 4). In the analysis, we found that DBSLMM and lassosum have the lowest computational cost both in terms of memory usage and in terms of computing time. Specifically, in the HM3 SNP set, it took the two methods an average of 18.97 and 17.49 min, with 0.17 and 3.69 Gb memory, to analyze the two traits, respectively. In contrast, NPS and PRS-CS are computationally slow (846.29 and 6004.11 min) but have low memory requirement (0.87 and 0.93 Gb). CT/SCT, LDpred2-auto, LDpred2-inf and SBLUP are computationally reasonably

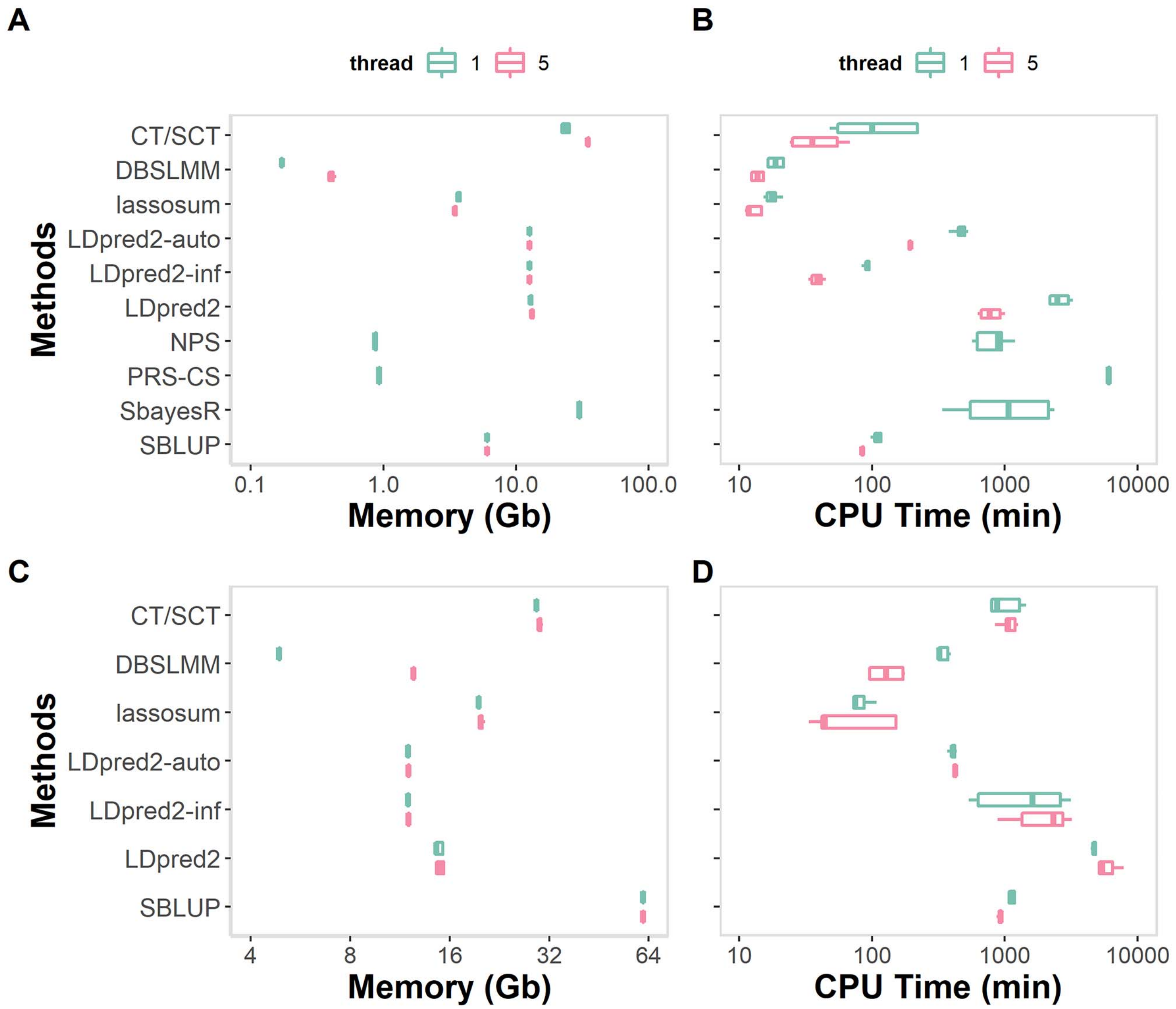

Figure 4. Physical memory usage and computing time for different PGS methods. Compared methods include CT/SCT, DBSLMM, lassosum, LDpred2-auto, LDpred2-inf, LDpred2 that include both LDpred2-nosp and LDpred2-sp, NPS, PRS-CS, SbayesR and SBLUP. Memory usage (**A**, **C**; in Gb) and computing time (**B**, **D**; in min) for different methods based on two SNP sets: the HM3 SNP set (**A**, **B**) and the BiLEVE SNP set (**C**, **D**). We examined two thread settings where we used either one CPU thread for computation (aquamarine) or five CPU threads for computation (peach). Comparison was performed based on Intel Xeon CPU E5–2683 2.00-GHz processors. Note that the x-axis for all panels is on log-scale.

fast (131.45, 467.26, 91.87 and 110.11 min) but require a reasonably large amount of memory (23.65, 12.61, 12.61 and 6.04 Gb). LDpred2 and SbayesR are both computationally slow (2608.11 and 1336.14 min) and require a large memory (12.88 and 30.02 Gb). In the BiLEVE SNP set, it took DBSLMM and lassosum an average of 341.35 and 82.25 min, with 4.88 and 19.39 Gb memory requirement, to analyze the two traits, respectively. The computing time and memory cost of LDpred2-auto (407.44 min and 11.97 Gb) is comparable with the two. The other methods, including CT/SCT (1040.15 min and 29.55 Gb), LDpred2-inf (1716.14 min and 11.97 Gb), LDpred2 (4720.51 min and 14.82 Gb) and SBLUP (1131.03 min and 61.52 Gb), incur substantially larger computing cost. Note that the three remaining methods (NPS, PRS-CS and SbayesR) cannot be applied to the BiLEVE SNP set as they require more than 64 Gb memory and/or take longer than 3 days without the ability to carry out computation in parallel based on individual chromosomes.

Importantly, five of the PGS software, including CT/SCT, DBSLMM, lassosum, LDpred2-auto, LDpred2-inf, LDpred2-nosp/sp and SBLUP, are capable of making use of the multithreading computing environment to improve computing speed further. In HM3 SNP set, by using five threads, the eight methods improve computing speed by an average of 119.53%, though with an average of 25.60% increase in memory requirement. In the BiLEVE SNP set, by using five threads, they improve computing speed by an average of 22.44%, though with an average of 26.53% increase in memory requirement.

PGS web server

We have created a PGS web server where users can construct PGS for their own applications. The server currently hosts the 12 PGS methods compared in the present study. With the server, users can fit any PGS method in a training data, tune its hyper-parameters in a validation data if required and output the inferred SNP weights for PGS construction. Users can also compute R^2 in a test data using the inferred SNP weights. The server only requires summary statistics and a reference LD panel as input. It is currently designed with the ability for efficient parallelization of computationally intensive tasks.

The PGS web server consists of four technical components: the Nginx proxy, the web frontend application, the web backend application and the PGS computing application. The Nginx proxy accepts HTTP requests from the website and forwards these requests to the web frontend graphical user interface (GUI) and/or backend application based on URL routing rules. The web frontend is designed as a web single page application (SPA) and is implemented based on the open-source frameworks React and AntDesign. The web frontend GUI facilitates multiple tasks that include file uploading, model parameter set up and results reporting. Specifically, the home page of the PGS webserver provides a procedure overview for PGS construction and validation. The navigation menu on the homepage consists of multiple items that allows users to either choose PGS methods for model fitting or computing R^2 in the test data with the estimated SNP weights. The web frontend communicates with the Java backend server by asynchronous HTTP requests (AJAX) with JSON as an interchange format. All transmissions between the frontend and backend are encrypted using secure socket layer (SSL). The Java backend application is implemented based on Spring Boot and RESTful web framework. The backend application receives AJAX requests from web frontend, validates input files and computation parameters and performs PGS modeling fitting or R^2 computation. The PGS server treats shell scripts as glue logic to connect different PGS methods, implemented by either C/C++, R or Python, with the web backend.

The PGS web server carries out PGS workflow in three separate steps (Figure 5). First, the server requires users to upload GWAS summary statistics in the GEMMA file format. To improve uploading efficiency for large files, the frontend web application automatically slices the uploading file into small segments, each of about 20 Mb in size. The web server displays a progress bar on the percentage of uploaded file during the uploading process. After file uploading, the PGS server relies on backend Java service application to check the updated files for basic file formatting such as filename extension, column number, separator of each column and computation parameters. The user will receive a format error if file check fails.

Second, the user choses one of the 12 PGS methods for model fitting. A dropdown manual is displayed, allowing user to select parameter initialization options for some of the PGS methods (CT, DBSLMM, LDpred2 and SCT). Based on the selected PGS method, the PGS server will register and allocate computing nodes to the desired analytic task and relies on shell scripts to call either the R function or the Python code to fit the corresponding PGS method. In the fitting process, PGS server manages and monitors the computation progress including node configuration through Linux shell scripts. The data processing status is monitored during computation via the output log file.

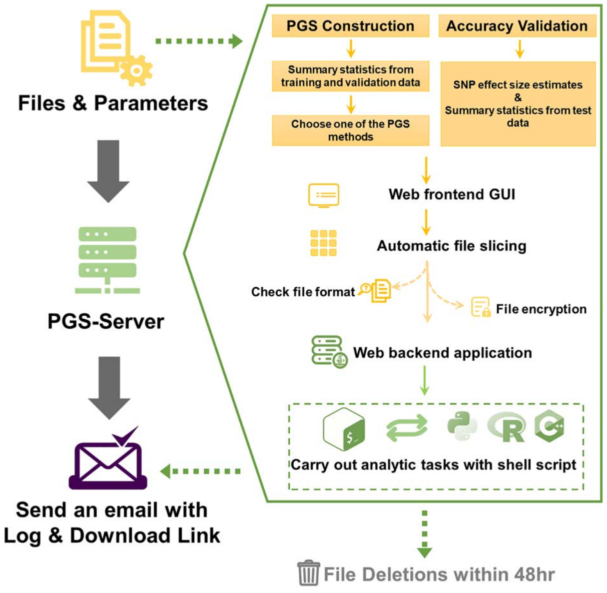
Files & Parameters
PGS-Server
Send an email with
Log & Download Link
PGS Construction
Accuracy Validation
Summary statistics from training and validation data
Choose one of the PGS methods
SNP effect size estimates & Summary statistics from test data
Web frontend GUI
Automatic file slicing
Check file format
File encryption
Web backend application
Carry out analytic tasks with shell script
File Deletions within 48hr

Figure 5. Analytic workflow of the PGS webserver. The PGS webserver carries out two distinct analytic tasks: PGS construction and accuracy evaluation. For the task of PGS construction, the users are required to upload the GWAS summary statistics for the training data, and if needed, the validation data. The users will also have the option to choose one of the 12 PGS methods for PGS construction. For the task of accuracy evaluation, the users are required to upload the estimated SNP effect sizes along with the GWAS summary statistics for the test data. The webserver will carry out accuracy evaluation by computing a prediction R^2 in the test data. For both tasks, files are uploaded in small segments to allow for efficient uploading. The webserver performs basic file format checking and calls the web backend application if the files pass the format check. The web backend application will then use the corresponding shell scripts to carry out the desired analytic task. At the end of the task, the webserver will send out an email with an attached log file and a download link for the user to retrieve their results. All uploaded files will be deleted within 48 hours after the email notification.

Finally, once the PGS method fitting jobs are finished, the shell script will return a code to the web frontend to report service success. In addition, an email will be sent to the user, with a link for downloading the log summary file and the output file, which contains either the effect size estimates (for PGS model fitting) or the test R^2 (for PGS construction and validation). In the workflow, PGS server employs two steps to ensure data security: the input file and estimation results are encrypted on the fly using a one-time password; and all input files, result files and final reports are deleted within 48 h after sending out the email notification.

Overall, we hope the PGS webserver serves as a useful and important tool for practitioners to perform PGS analysis in their own applications.

Discussion

We have presented a comprehensive benchmarking study on 12 PGS methods plus an aggregation approach through UKB internal validations as well as cross-ancestry and cross-study external validations. We have compared the accuracy of PGS methods for both quantitative traits and binary phenotypes and recorded their computational cost in terms of computing time and memory requirement. We show that a key determent of PGS method performance is the modeling assumption on the distribution of SNP effect sizes. Indeed, our results suggest that polygenic methods with local shrinkage on the SNP effect sizes often achieve higher accuracy than global shrinkage methods, sparse methods as well as *ad hoc* algorithm based PGS construction approaches. Besides method comparison, we have presented a simple aggregating approach to combine the 12 PGS from different methods into a single PGS that achieves superior prediction accuracy across a wide range of traits in both internal and external validations. In addition, we have presented a PGS webserver to facilitate the adaptation of different PGS methods for routine analysis. We hope that the detailed comparison of these state-of-the-art PGS methods, the aggregation approach, the PGS server and our recommendations for choosing PGS methods (Figure 6) can serve as a useful guideline and an analytic platform for practitioners.

The present study compared 12 PGS methods plus an aggregation approach on 25 quantitative traits and 25 binary traits from 50 datasets in the internal validation and 137 datasets in the external validation. The number of PGS methods and the number of traits examined in the present study are much larger than those examined in early comparable studies. For example, Pain *et al*. compared eight PGS methods for seven quantitative and eight binary traits [40]. Ni *et al*. compared 10 PGS methods for two binary traits [42]. Privé *et al*. compared eight PGS methods for eight diseases [33]. Although our study examined a large number of PGS methods across a large number of traits among these studies, we did find that some of the results we obtain share certain consistency with the three previous comparative studies. For example, the three previous studies found that the performance of LDpred2 is better than lassosum, PRS-CS and SBLUP. We find similar results: the relative performance of LDpred2-nosp and LDpred2-sp in the 25 quantitative traits on average is 0.93 and 0.92, respectively, which is better than that of the other three methods (0.90, 0.58, 0.70, respectively). As another example, the previous studies found that the tuning version of PRS-CS and LDpred2 performs better than automatic version PRS-CS$_{\text{auto}}$ and LDpred2-inf, respectively. We also found that the tuning version of DBSLMM is on average 1.13% (median = 1.14%) better than DBSLMM$_{\text{auto}}$, whereas the tuning version of PRS-CS is on average 12.29% (median = 3.48%) better than PRS-CS$_{\text{auto}}$ across 50 traits. Nevertheless, some of our results are different from the previous three studies. For example, although the previous studies found the performance of CT generally to be the worst, we found that its performance is on the low end but not always the worst. The moderate performance of CT observed in the present study is presumably because we used 1400 hyper-parameters combination for CT, which represents a much larger parameter space than previously explored.

In the present study, we have primarily focused on using the default software settings when applying different PGS methods. We acknowledge, however, that modifying the software setting for certain methods on certain data types may help improve performance. We have primarily focused on comparing the prediction accuracy of different PGS methods. We note that PGS methods have also been used for many downstream analytic tasks other than prediction in GWASs. For example, PGS methods have been widely used for risk stratification, phenome-wide association studies, Mendelian randomization and transcriptome association studies (TWASs; [72, 73]). Evaluating the performance of PGS methods for other analytic tasks is an important future research direction.

We have primarily focused on modeling quantitate traits and binary traits. Time-to-event data is another important data type that is becoming common to genetic study of human disease with Biobank scale data. The standard approach for analyzing time-to-event data is the Cox regression model, also known as the proportional hazards model, which examines the association between a time-to-event outcome variable and an exposure variable of interest. Although a typical Cox regression model accommodates a handful of exposure variables, recent studies have extended the Cox regression model to incorporate a large number of exposure variables by placing lasso, Elastic Net or other penalties on the model coefficients [74]. These penalized Cox regression models can be directly applied to the setting of PGS, where we can treat the SNPs as the exposure variables and the time-to-event as the outcome variable. The estimated SNP coefficients from such penalized Cox regression model can be used to construct a predictor for the latent proportional hazards, which can be further used to predict the time-to-event. Certainly, fitting the penalized Cox regression models is computationally demanding. For example, the packages glmnet [75], penalized [76], coxpath [77] and glcoxph [78] use different fitting strategies but all require exceedingly large amount of RAM memory, especially for fitting biobank scale genotype data. Consequently, several recent PGS analysis of time-to-event data directly fitted a Cox regression model on one SNP at a time and constructed a polygenic hazard score by summing the coefficient estimates from the fitted SNPs [79]. However, fitting one SNP at a time ignores the correlation among SNPs due to LD. Importantly, Li *et al*. recently proposed a new approach, snpnet-Cox, to solve L^1 regularized Cox regression for large-scale and high-dimensional data that do not fit in the memory

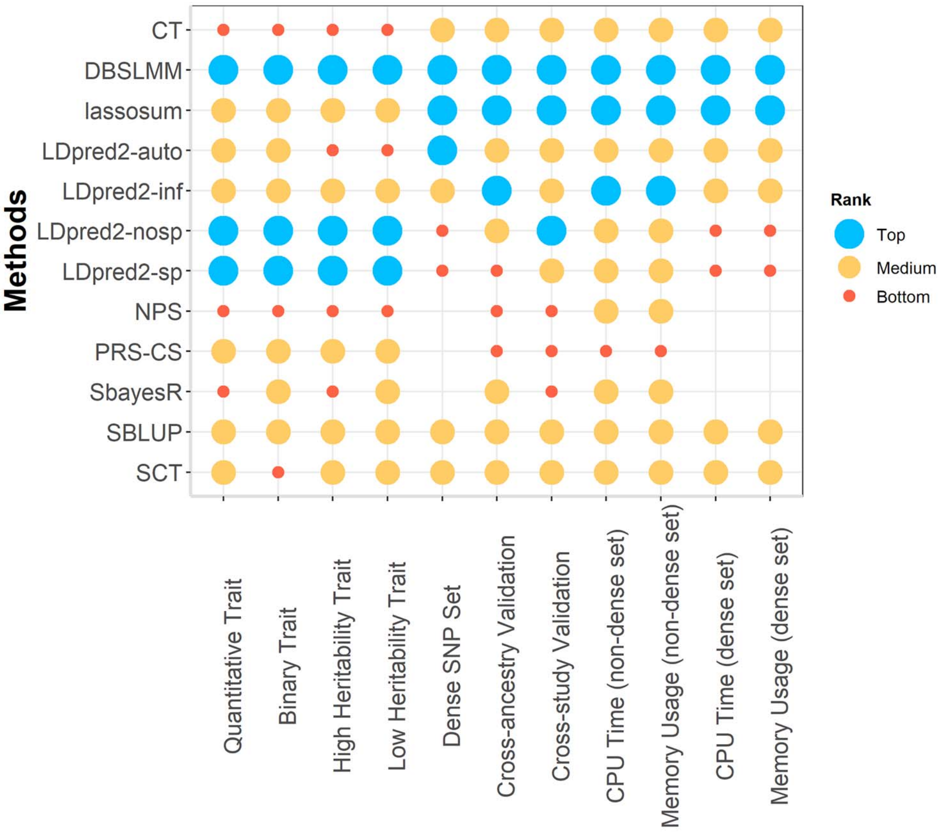

Figure 6. Practical guideline for choosing PGS methods. Compared methods include CT, DBSLMM, lassosum, LDpred2-auto, LDpred2-inf, LDpred2-nosp, LDpred2-sp, NPS, PRS-CS, SbayesR, SBLUP and SCT. Methods (y-axis) are first ranked based on their average performance across each examined criterion (x-axis) and are then categorized into three groups that include a top performance group (top three methods; deep sky blue colored large circles), a medium performance group (navajo white colored medium circles) and a bottom performance group (bottom three methods; tomato colored small circles). For the performance evaluated based on the dense SNP set, we categorized the top two methods into the top performance group and the bottom two methods into the poor performance group as three methods (NPS, PRS-CS and SbayesR) require too much computational resources and thus cannot be applied there.

and can be directly applied to UKB data [80]. Therefore, exploring similar computationally efficient ways to fit other types of penalized Cox regression models beyond L^1 regularization is an important future direction for constructing accurate PGS for time-to-event data.

We have primarily focused on evaluating model-based and non-model-based PGS methods that use only GWAS information. Several recent PGS methods have been developed to incorporate additional and external information beyond what is available in the GWAS data. Such external information can be either in the form of SNP functional annotations or in the form of other phenotypes in addition to the phenotype of interest. Specifically, several PGS methods have been developed to incorporate SNP functional annotations to improve phenotype prediction. For example, AnnoPred incorporates SNP functional annotations into the prior distribution of effect sizes based on a sparse modeling assumption on the SNP effect sizes [81]. LDpred-funct builds upon the LMM and models the SNP effect size variance as a function of SNP annotations [82]. In addition, several PGS methods have been developed to incorporate pleiotropic information that characterizes SNP effects similarity across multiple correlated traits to improve SNP effect size estimation on the trait of interest. PGS methods that take advantage of pleiotropy are often based on the multivariate LMM [83], also known as the MT-BLUP in prediction settings. Example PGS methods that make use of pleiotropic information include MTGBLUP [84] and wMT-SBLUP [85]. Finally, other methods, such as PleioPred [86], also incorporate SNP functional annotations into pleiotropic modeling. Incorporating external information can often improve the accuracy of PGS. Therefore, evaluating additional PGS methods that incorporate external information and evaluating the benefits of the different sources of external information for improving prediction performance is another important future direction.

Key Points

- We systemically compared the prediction performance of existing PGS construction methods using internal and external validation data.
- We presented a simple aggregation strategy that combines multiple PGS from different methods to take advantage of their distinct benefits to achieve stable and superior prediction performance.

- We developed a PGS webserver that allows users to upload their own GWAS summary statistics and choose different PGS methods to fit the data directly on the server.

Supplementary data

Supplementary data are available online at *Briefings in Bioinformatics*.

Authors' contributions

S.Y. and X.Z. conceived the study and designed the experiments. S.Y. performed all analyses and developed the web server. S.Y. and X.Z. wrote the manuscript. All authors reviewed and approved this version of the manuscript.

Conflict of interest

The authors declare that they have no competing interests.

Data availability

The dataset(s) supporting the conclusions of this article is(are) available in the UK Biobank resource [http://www.ukbiobank.ac.uk] under Application Number 67665. UK Biobank was established by the Wellcome Trust medical charity, Medical Research Council, Department of Health, Scottish Government and the Northwest Regional Development Agency. It has also had funding from the Welsh Assembly Government, British Heart Foundation and Diabetes UK. All genome-wide association summary statistics used in this study are publicly available.

A web server code for PGS-Server is available on github repository (https://github.com/biostat0903/PGS-Server). The web server is available on (http://www.pgs-server.com/).

Funding

This work was supported by the Natural Science Foundation of China (81703321 to S.Y.); and the University of Michigan (to X.Z.).

References

1. Sakaue S, Kanai M, Karjalainen J, *et al.* Trans-biobank analysis with 676,000 individuals elucidates the association of polygenic risk scores of complex traits with human lifespan. *Nat Med* 2020;**26**:542–8.
2. Zijie Zhao JS, Wang T. Qiongshi Lu. Polygenic risk scores: effect estimation and model optimization. *Quant Biol* 2021;**9**:133–40.
3. Ma Y, Zhou X. Genetic prediction of complex traits with polygenic scores: a statistical review. *Trends Genet* 2021;**37**:995–1011.
4. Shungin D, Winkler TW, Croteau-Chonka DC, *et al.* New genetic loci link adipose and insulin biology to body fat distribution. *Nature* 2015;**518**:187–96.
5. Purcell SM, Wray NR, Stone JL, *et al.* Common polygenic variation contributes to risk of schizophrenia and bipolar disorder. *Nature* 2009;**460**:748–52.
6. Visscher PM, Wray NR, Zhang Q, *et al.* 10 years of GWAS discovery: biology, function, and translation. *Am J Hum Genetics* 2017;**101**:5–22.
7. Wang Y, Guo J, Ni G, *et al.* Theoretical and empirical quantification of the accuracy of polygenic scores in ancestry divergent populations. *Nat Commun* 2020;**11**:3865.
8. Elliott J, Bodinier B, Bond TA, *et al.* Predictive accuracy of a polygenic risk score–enhanced prediction model vs a clinical risk score for coronary artery disease. *JAMA* 2020;**323**:636–45.
9. Forgetta V, Keller-Baruch J, Forest M, *et al.* Development of a polygenic risk score to improve screening for fracture risk: a genetic risk prediction study. *PLoS Med* 2020;**17**:e1003152.
10. Marston NA, Kamanu FK, Nordio F, *et al.* Predicting benefit from evolocumab therapy in patients with atherosclerotic disease using a genetic risk score. *Circulation* 2020;**141**:616–23.
11. Moll M, Sakornsakolpat P, Shrine N, *et al.* Chronic obstructive pulmonary disease and related phenotypes: polygenic risk scores in population-based and case-control cohorts. *Lancet Respir Med* 2020;**8**:696–708.
12. Perkins DO, Loohuis LO, Barbee J, *et al.* Polygenic risk score contribution to psychosis prediction in a target population of persons at clinical high risk. *Am J Psychiatry* 2020;**177**:155–63.
13. Dai J, Lv J, Zhu M, *et al.* Identification of risk loci and a polygenic risk score for lung cancer: a large-scale prospective cohort study in Chinese populations. *Lancet Respir Med* 2019;**7**:881–91.
14. Cases in Precision Medicine. The role of polygenic risk scores in breast cancer risk assessment. *Ann Intern Med* **174**:408–12.
15. Meisner A, Kundu P, Zhang YD, *et al.* Combined utility of 25 disease and risk factor polygenic risk scores for stratifying risk of all-cause mortality. *Am J Hum Genet* 2020;**107**:418–31.
16. Khera AV, Chaffin M, Aragam KG, *et al.* Genome-wide polygenic scores for common diseases identify individuals with risk equivalent to monogenic mutations. *Nat Genet* 2018;**50**:1219–24.
17. Thomas M, Sakoda LC, Hoffmeister M, *et al.* Genome-wide modeling of polygenic risk score in colorectal cancer risk. *Am J Hum Genet* 2020;**107**:432–44.
18. Panyard DJ, Deming YK, Darst BF, *et al.* Liver-specific polygenic risk score is more strongly associated than genome-wide score with Alzheimer's disease diagnosis in a case-control analysis. *medRxiv* 2021; 2021.2004.2029.21256279.
19. Beesley LJ, Salvatore M, Fritsche LG, *et al.* The emerging landscape of health research based on biobanks linked to electronic health records: existing resources, statistical challenges, and potential opportunities. *Stat Med* 2020;**39**:773–800.
20. Sudlow C, Gallacher J, Allen N, *et al.* UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Med* 2015;**12**:e1001779.
21. Nagai A, Hirata M, Kamatani Y, *et al.* Overview of the BioBank Japan project: study design and profile. *J Epidemiol* 2017;**27**:S2–8.
22. Chen Z, Chen J, Collins R, *et al.* China Kadoorie Biobank of 0.5 million people: survey methods, baseline characteristics and long-term follow-up. *Int J Epidemiol* 2011;**40**:1652–66.
23. Locke AE, Steinberg KM, Chiang CWK, *et al.* Exome sequencing of Finnish isolates enhances rare-variant association power. *Nature* 2019;**572**:323–8.
24. Denny JC, Rutter JL, Goldstein DB, *et al.* The "All of Us" Research Program. *N Engl J Med* 2019;**381**:668–76.
25. Li YR, Keating BJ. Trans-ethnic genome-wide association studies: advantages and challenges of mapping in diverse populations. *Genome Med* 2014;**6**:91.

26. Chen M-H, Raffield LM, Mousas A, *et al.* Trans-ethnic and ancestry-specific blood-cell genetics in 746,667 individuals from 5 global populations. *Cell* 2020;**182**:1198–1213.e1114.
27. Yang S, Zhou X. Accurate and scalable construction of polygenic scores in large biobank data sets. *Am J Hum Genet* 2020;**106**: 679–93.
28. Duncan L, Shen H, Gelaye B, *et al.* Analysis of polygenic risk score usage and performance in diverse human populations. *Nat Commun* 2019;**10**:3328.
29. Cai M, Xiao J, Zhang S, *et al.* A unified framework for cross-population trait prediction by leveraging the genetic correlation of polygenic traits. *Am J Hum Genet* 2021;**108**:632–55.
30. Choi SW, Mak TS-H, O'Reilly PF. Tutorial: a guide to performing polygenic risk score analyses. *Nat Protoc* 2020;**15**:2759–72.
31. Zhou X, Carbonetto P, Stephens M. Polygenic Modeling with Bayesian sparse linear mixed models. *PLoS Genet* 2013; **9**:e1003264.
32. Zhao Z, Yi Y, Song J, *et al.* PUMAS: fine-tuning polygenic risk scores with GWAS summary statistics. *Genome Biol* 2021;**22**:257.
33. Privé F, Arbel J, Vilhjálmsson BJ. LDpred2: better, faster, stronger. *Bioinformatics* 2020;**36**:5424–31.
34. Robinson MR, Kleinman A, Graff M, *et al.* Genetic evidence of assortative mating in humans. *Nat Hum Behav* 2017;**1**: 0016.
35. Mak TSH, Porsch RM, Choi SW, *et al.* Polygenic scores via penalized regression on summary statistics. *Genet Epidemiol* 2017;**41**: 469–80.
36. Zeng P, Zhou X. Non-parametric genetic prediction of complex traits with latent Dirichlet process regression models. *Nat Commun* 2017;**8**:456.
37. Ge T, Chen C-Y, Ni Y, *et al.* Polygenic prediction via Bayesian regression and continuous shrinkage priors. *Nat Commun* 2019;**10**:1776.
38. Lloyd-Jones LR, Zeng J, Sidorenko J, *et al.* Improved polygenic prediction by Bayesian multiple regression on summary statistics. *Nat Commun* 2019;**10**:5086.
39. Privé F, Vilhjálmsson BJ, Aschard H, *et al.* Making the most of clumping and thresholding for polygenic scores. *Am J Hum Genet* 2019;**105**:1213–21.
40. Pain O, Glanville KP, Hagenaars SP, *et al.* Evaluation of polygenic prediction methodology within a reference-standardized framework. *PLoS Genet* 2021;**17**:e1009021.
41. Kulm S, Marderstein A, Mezey J, *et al.* A systematic framework for assessing the clinical impact of polygenic risk scores. *medRxiv* 2021:2020.2004.2006.20055574.
42. Ni G, Zeng J, Revez JA, *et al.* A comparison of ten polygenic score methods for psychiatric disorders applied across multiple cohorts. *Biol Psychiatry* 2021;**90**:611–20.
43. Martin AR, Gignoux CR, Walters RK, *et al.* Human demographic history impacts genetic risk prediction across diverse populations. *Am J Hum Genet* 2017;**100**:635–49.
44. Privé F, Aschard H, Ziyatdinov A, *et al.* Efficient analysis of large-scale genome-wide data with two R packages: bigstatsr and bigsnpr. *Bioinformatics* 2018;**34**:2781–7.
45. Chun S, Imakaev M, Hui D, *et al.* Non-parametric polygenic risk prediction via partitioned GWAS summary statistics. *Am J Hum Genet* 2020;**107**:46–59.
46. Berisa T, Pickrell JK. Approximately independent linkage disequilibrium blocks in human populations. *Bioinformatics* 2015;**32**: 283–5.
47. Wen X, Stephens M. Using linear predictors to impute allele frequencies from summary or pooled genotype data. *Ann Appl Stat* 2010;**4**:1158–1182, 1125.
48. Zhou X, Stephens M. Genome-wide efficient mixed-model analysis for association studies. *Nat Genet* 2012;**44**:821–4.
49. Chang CC, Chow CC, Tellier LC, *et al.* Second-generation PLINK: rising to the challenge of larger and richer datasets. *GigaScience* 2015;**4**:s13742–13015–10047–13748.
50. Fisher RA. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika* 1915;**10**:507–21.
51. Akiyama M, Ishigaki K, Sakaue S, *et al.* Characterizing rare and low-frequency height-associated variants in the Japanese population. *Nat Commun* 2019;**10**:4393.
52. Kanai M, Akiyama M, Takahashi A, *et al.* Genetic analysis of quantitative traits in the Japanese population links cell types to complex human diseases. *Nat Genet* 2018;**50**:390–400.
53. Akiyama M, Okada Y, Kanai M, *et al.* Genome-wide association study identifies 112 new loci for body mass index in the Japanese population. *Nat Genet* 2017;**49**:1458–67.
54. Horikoshi M, Day FR, Akiyama M, *et al.* Elucidating the genetic architecture of reproductive ageing in the Japanese population. *Nat Commun* 2018;**9**:1977.
55. Leslie R, O'Donnell CJ, Johnson AD. GRASP: analysis of genotype–phenotype results from 1390 genome-wide association studies and corresponding open access database. *Bioinformatics* 2014;**30**:i185–94.
56. Watanabe K, Stringer S, Frei O, *et al.* A global overview of pleiotropy and genetic architecture in complex traits. *Nat Genet* 2019;**51**:1339–48.
57. Wood AR, Esko T, Yang J, *et al.* Defining the role of common variation in the genomic and biological architecture of adult human height. *Nat Genet* 2014;**46**:1173–86.
58. Medina-Gomez C, Kemp JP, Trajanoska K, *et al.* Life-course genome-wide association study meta-analysis of total body BMD and assessment of age-specific effects. *Am J Hum Genet* 2018;**102**:88–102.
59. Willer CJ, Schmidt EM, Sengupta S, *et al.* Discovery and refinement of loci associated with lipid levels. *Nat Genet* 2013;**45**: 1274–83.
60. Kathiresan S, Willer CJ, Peloso GM, *et al.* Common variants at 30 loci contribute to polygenic dyslipidemia. *Nat Genet* 2009;**41**: 56–65.
61. Kettunen J, Demirkan A, Würtz P, *et al.* Genome-wide study for circulating metabolites identifies 62 loci and reveals novel systemic effects of LPA. *Nat Commun* 2016;**7**:11122.
62. Ferreira MAR, Hottenga J-J, Warrington NM, *et al.* Sequence variants in three loci influence monocyte counts and erythrocyte volume. *Am J Hum Genet* 2009;**85**:745–9.
63. Locke AE, Kahali B, Berndt SI, *et al.* Genetic studies of body mass index yield new insights for obesity biology. *Nature* 2015;**518**: 197–206.
64. Perry JRB, Day F, Elks CE, *et al.* Parent-of-origin-specific allelic associations among 106 genomic loci for age at menarche. *Nature* 2014;**514**:92–7.
65. Day FR, Thompson DJ, Helgason H, *et al.* Genomic analyses identify hundreds of variants associated with age at menarche and support a role for puberty timing in cancer risk. *Nat Genet* 2017;**49**:834–41.
66. Shrine N, Guyatt AL, Erzurumluoglu AM, *et al.* New genetic signals for lung function highlight pathways and chronic obstructive pulmonary disease associations across multiple ancestries. *Nat Genet* 2019;**51**:481–93.
67. Warrington NM, Beaumont RN, Horikoshi M, *et al.* Maternal and fetal genetic effects on birth weight and their relevance to cardio-metabolic risk factors. *Nat Genet* 2019;**51**:804–14.

68. Horikoshi M, Yaghootkar H, Mook-Kanamori DO, *et al.* New loci associated with birth weight identify genetic links between intrauterine growth and adult height and metabolism. *Nat Genet* 2013;**45**:76–82.
69. Polson NG, Scott JG. Alternative global–local shrinkage priors using hypergeometric–beta mixtures. *Tech Rep* 2009–14. Duke University, Department of Statistical Science 2009.
70. Bycroft C, Freeman C, Petkova D, *et al.* The UK Biobank resource with deep phenotyping and genomic data. *Nature* 2018;**562**: 203–9.
71. Martin AR, Kanai M, Kamatani Y, *et al.* Clinical use of current polygenic risk scores may exacerbate health disparities. *Nat Genet* 2019;**51**:584–91.
72. Gusev A, Ko A, Shi H, *et al.* Integrative approaches for large-scale transcriptome-wide association studies. *Nat Genet* 2016;**48**: 245–52.
73. Daghlas I, Richmond RC, Lane JM, *et al.* Selection into shift work is influenced by educational attainment and body mass index: a Mendelian randomization study in the UK Biobank. *Int J Epidemiol* 2021;**50**:1229–40.
74. Qian J, Tanigawa Y, Du W, *et al.* A fast and scalable framework for large-scale and ultrahigh-dimensional sparse regression with application to the UK Biobank. *PLoS Genet* 2020;**16**:e1009141.
75. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* 2010;**33**:1.
76. Goeman JJ. L1 penalized estimation in the cox proportional hazards model. *Biom J* 2010;**52**:70–84.
77. Park MY, Hastie T. L1-regularization path algorithm for generalized linear models. *J R Stat Soc Series B Stat Methodology* 2007;**69**: 659–77.
78. Sohn I, Kim J, Jung S-H, *et al.* Gradient lasso for cox proportional hazards model. *Bioinformatics* 2009;**25**: 1775–81.
79. Liu G, Peng J, Liao Z, *et al.* Genome-wide survival study identifies a novel synaptic locus and polygenic score for cognitive progression in Parkinson's disease. *Nat Genet* 2021;**53**: 787–93.
80. Li R, Chang C, Justesen JM, *et al.* Fast Lasso method for large-scale and ultrahigh-dimensional cox model with applications to UK Biobank. *Biostatistics* 2020;kxaa038.
81. Hu Y, Lu Q, Powles R, *et al.* Leveraging functional annotations in genetic risk prediction for human complex diseases. *PLoS Comput Biol* 2017;**13**:e1005589.
82. Márquez-Luna C, Gazal S, Loh P-R, *et al.* Incorporating functional priors improves polygenic prediction accuracy in UK Biobank and 23andMe data sets. *Nat Commun* 2021;**12**: 6052.
83. Zhou X, Stephens M. Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nat Methods* 2014;**11**:407–9.
84. Maier R, Moser G, Chen G-B, *et al.* Joint analysis of psychiatric disorders increases accuracy of risk prediction for schizophrenia, bipolar disorder, and major depressive disorder. *Am J Hum Genet* 2015;**96**:283–94.
85. Maier RM, Zhu Z, Lee SH, *et al.* Improving genetic prediction by leveraging genetic correlations among human diseases and traits. *Nat Commun* 2018;**9**:989.
86. Hu Y, Lu Q, Liu W, *et al.* Joint modeling of genetically correlated diseases and functional annotations increases accuracy of polygenic risk prediction. *PLoS Genet* 2017;**13**: e1006836.