The American Journal of Human Genetics, Volume *109*

Supplemental information

Portability of 245 polygenic scores when derived from the UK Biobank and applied to 9 ancestry groups from the same cohort

Florian Privé, Hugues Aschard, Shai Carmi, Lasse Folkersen, Clive Hoggart, Paul F. O'Reilly, and Bjarni J. Vilhjálmsson

Supplementary Tables and Figures

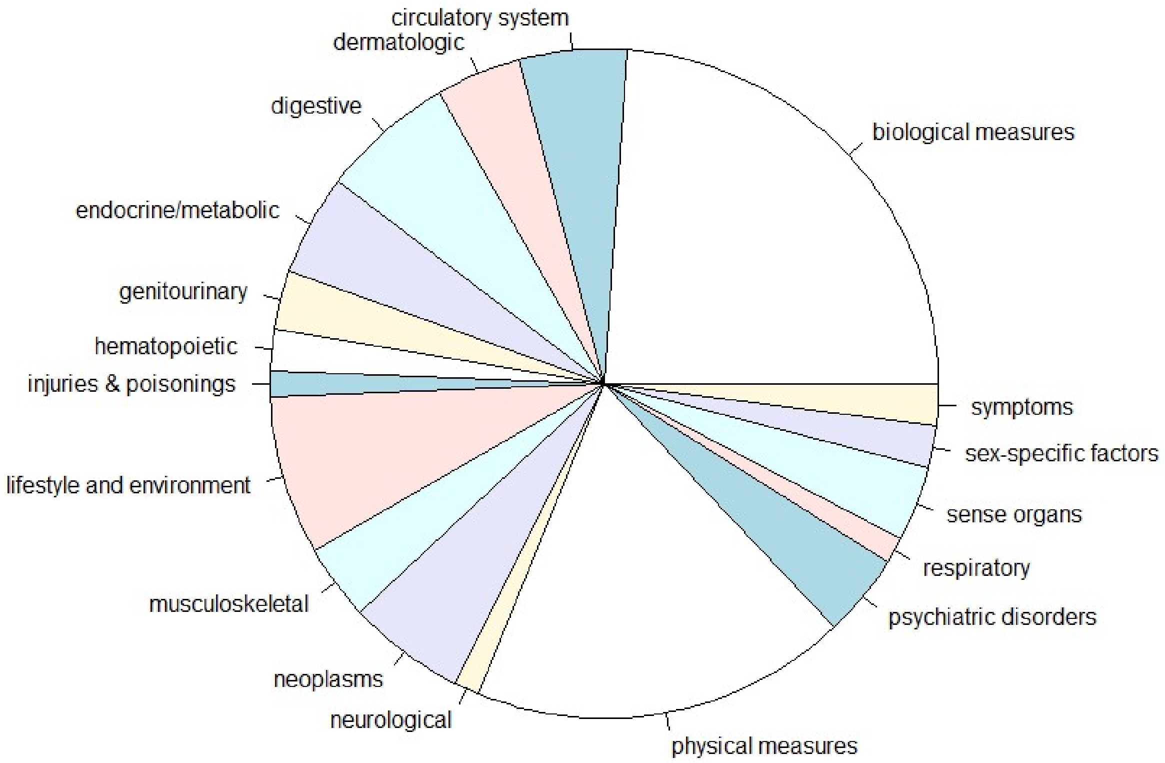
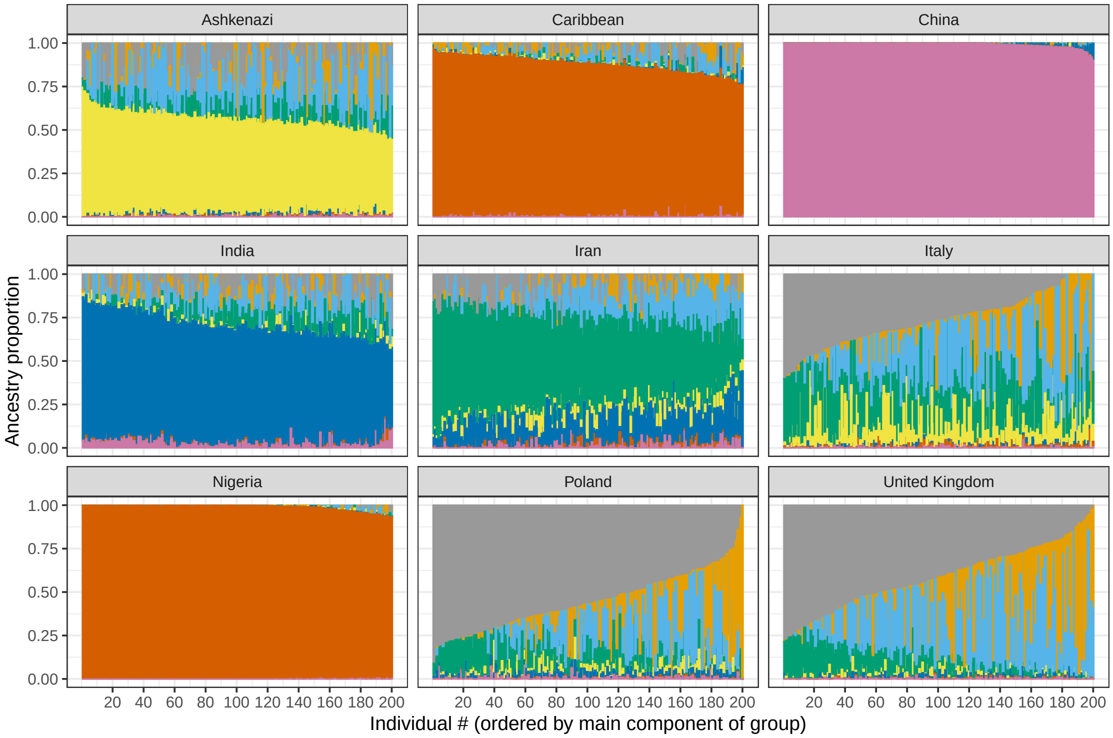
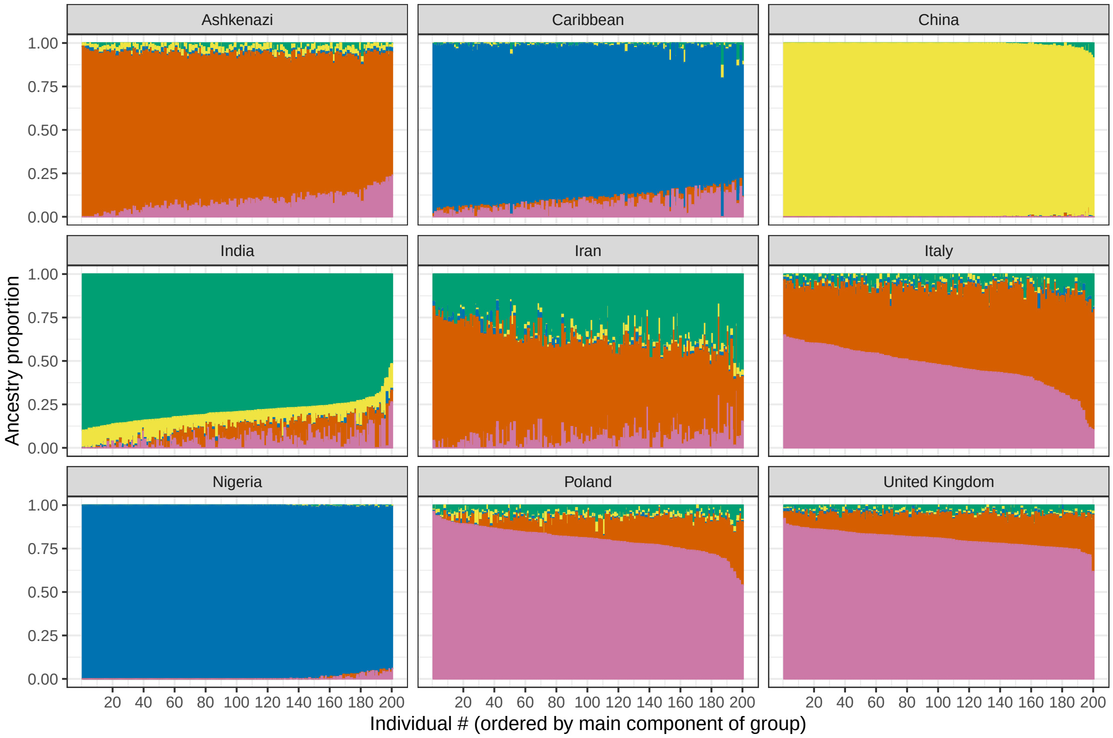


Figure S1: Pie chart of the categories of the 245 phenotypes used in this study. A full description of these phenotypes can be downloaded at `https://github.com/privefl/UKBB-PGS/blob/main/phenotype-description.xlsx`.



(a) with $K = 8$ components



(b) with $K = 5$ components

Figure S2: Results of running ADMIXTURE (Alexander *et al.* 2009) on 200 individuals from each of the nine ancestry groups we define in this study.

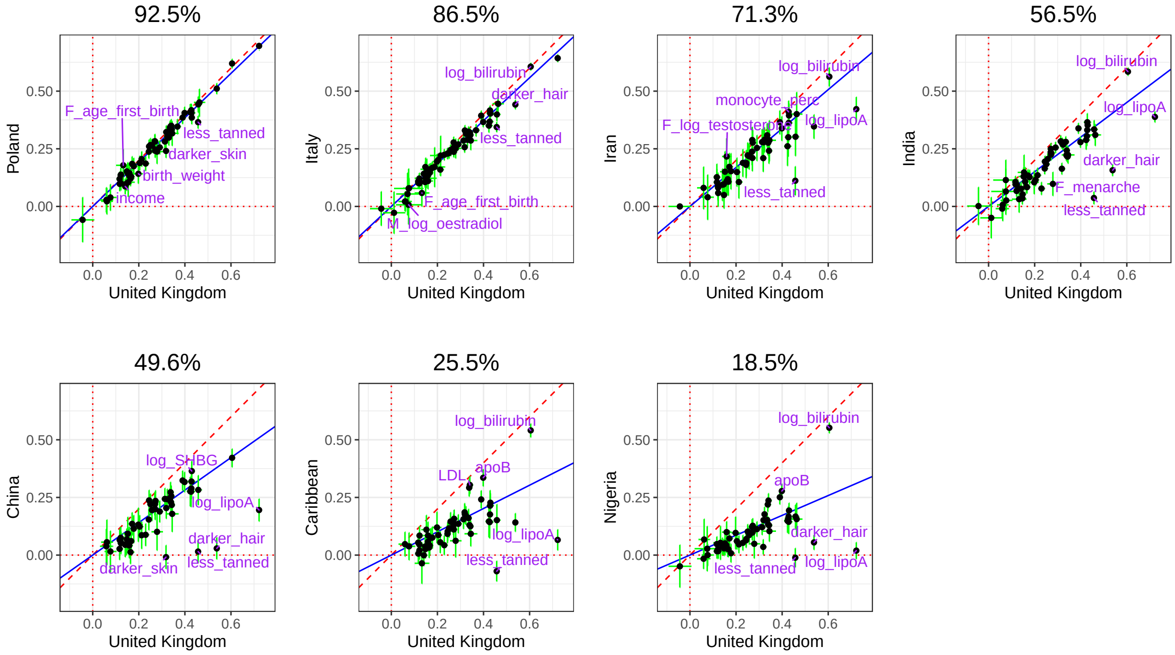


Figure S3: Partial correlation (and 95% CI) in the UK test set versus in a test set from another ancestry group. Each point represents a phenotype (only 83 of the continuous phenotypes here) and training has been performed with penalized regression on UK individuals (training 1 in table 1) and **genotyped** variants. The slope (in blue) is computed using Deming regression accounting for standard errors in both x and y, fixing the intercept at 0. The square of this slope is provided above each plot, which we report as the relative predictive performance compared to testing in UK.

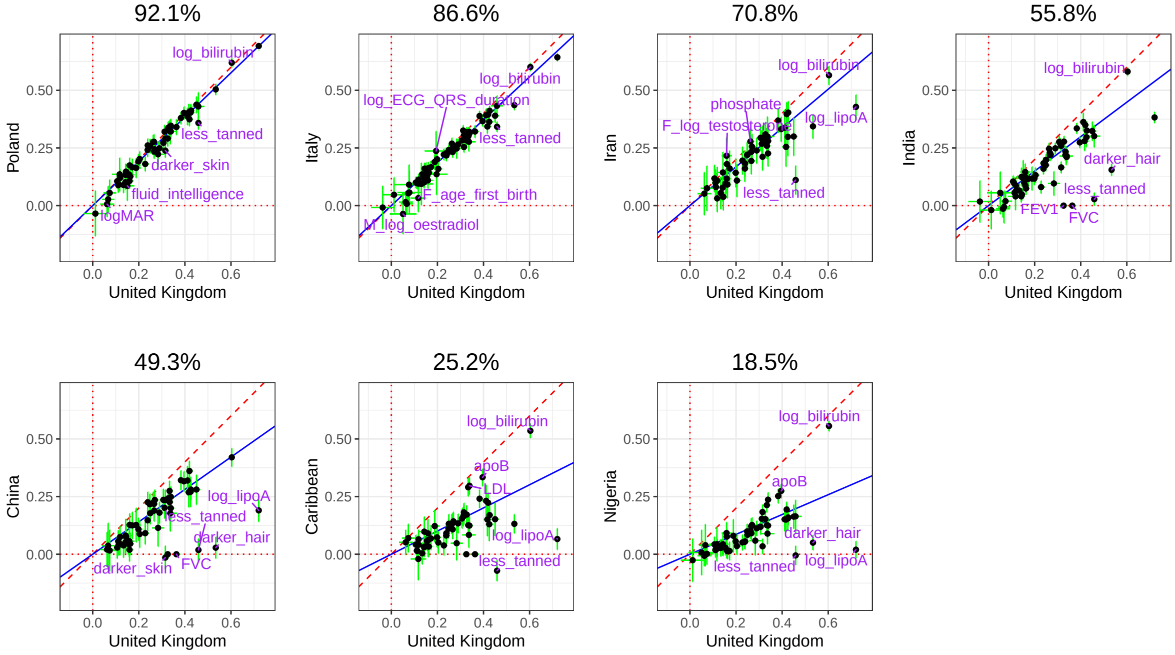


Figure S4: Results from identical analyses as in figure S3 except that we also remove third-degree relatives in the UK Biobank data we use (instead of second-degree and closer before).

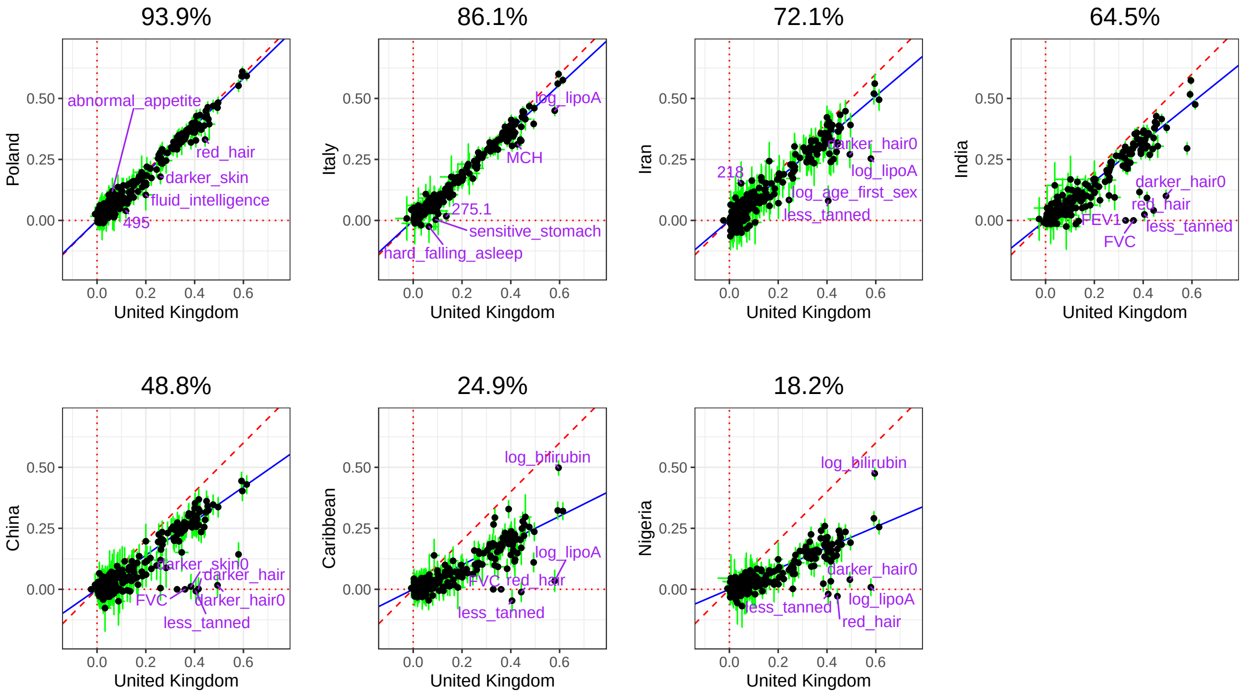


Figure S5: Partial correlation (and 95% CI) in the UK test set versus in a test set from another ancestry group. Each point represents a phenotype and training has been performed with **LDpred2-auto** on UK individuals (training 1 in table 1) and HapMap3 variants. The slope (in blue) is computed using Deming regression accounting for standard errors in both x and y, fixing the intercept at 0. The square of this slope is provided above each plot, which we report as the relative predictive performance compared to testing in UK.

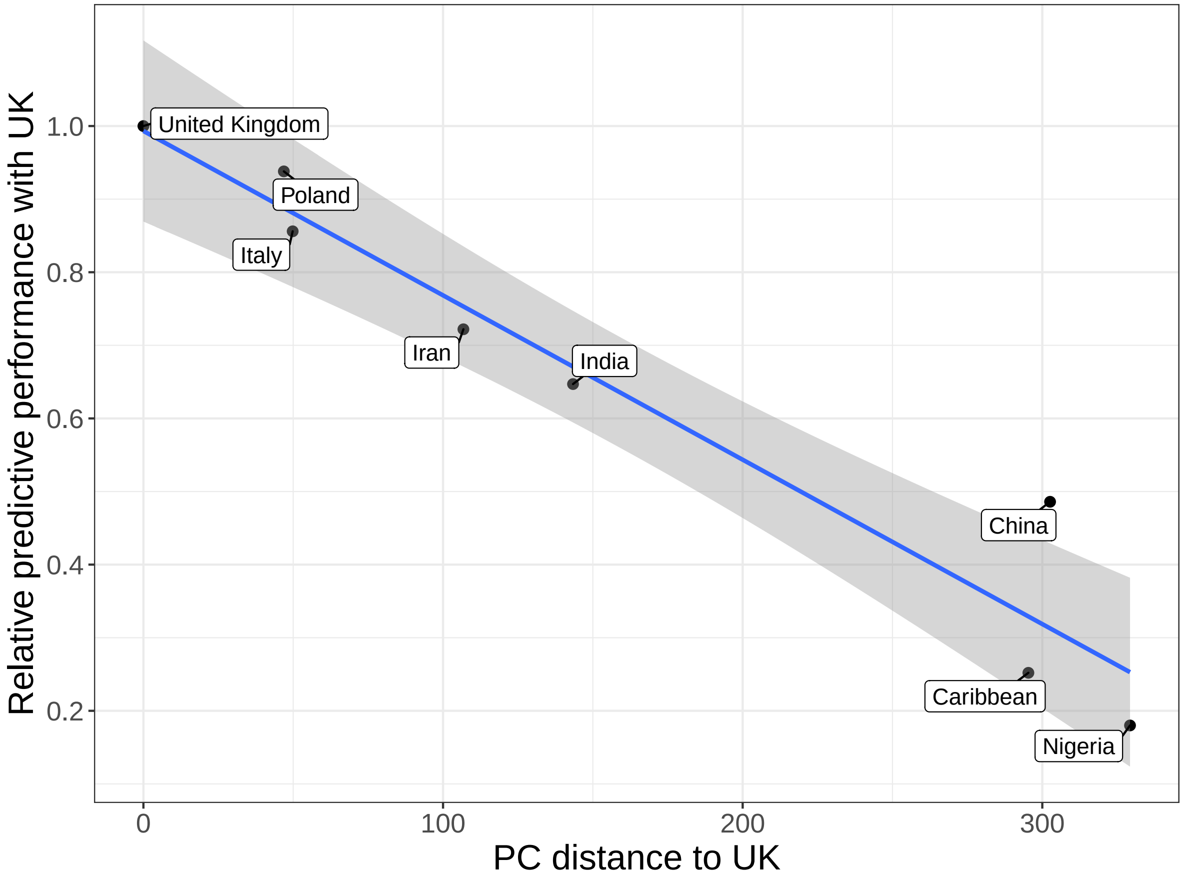


Figure S6: Relative predictive performance compared to the UK (ratio of variance explained in one group compared to in the UK group) versus PC distance from the UK. PCA is computed using individuals from test 1 (Table 1), and PC distances are computed using Euclidean distance between geometric medians of the first 32 PC scores of each ancestry group (shown in figure S7). Relative performance values are the ones reported in figure 2 of the main text. The slope and standard errors are computed internally by function `geom_smooth(method = "lm")` of R package ggplot2.

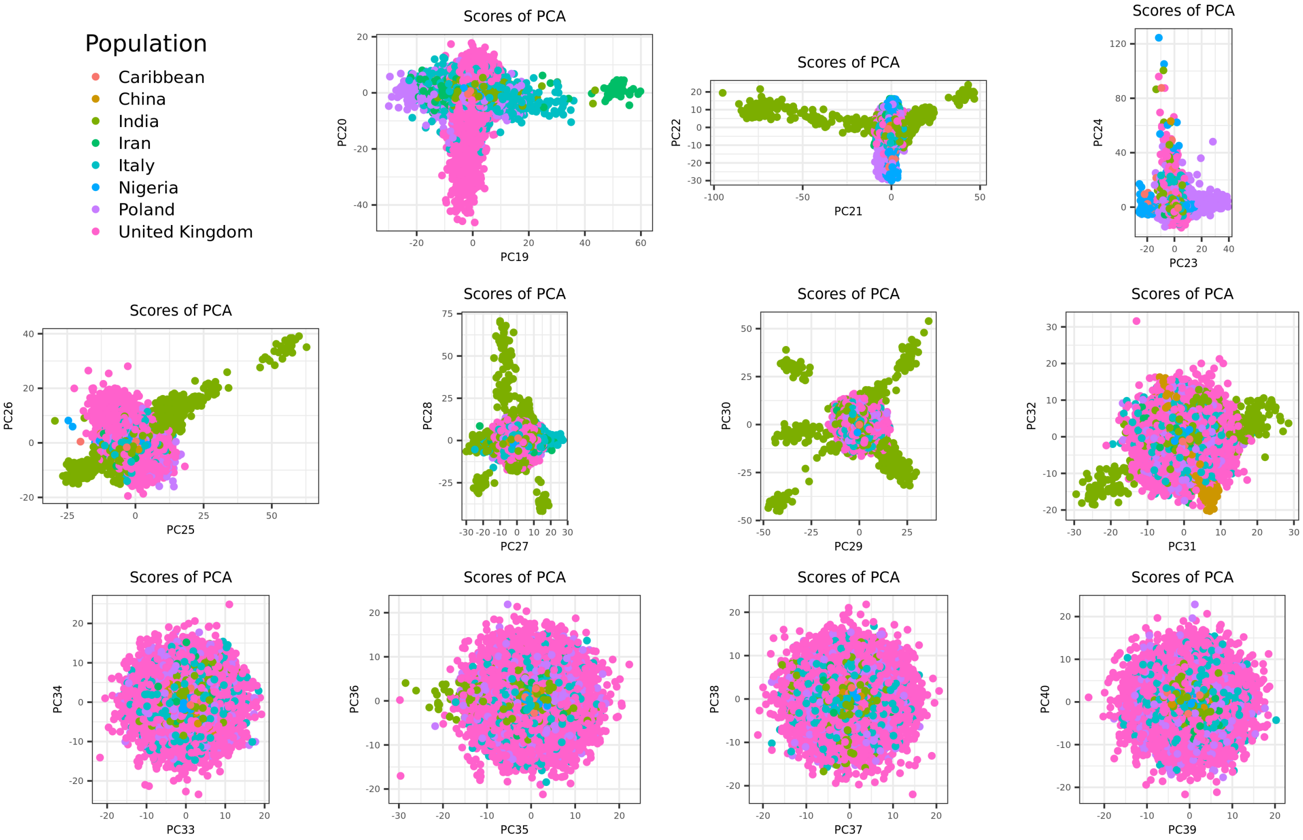


Figure S7: PC scores 19 to 40 when PCA is computed using individuals from test 1 (Table 1). PCs 19 to 32 visually capture some population structure, so we use first 32 PCs when computing the PC distances.

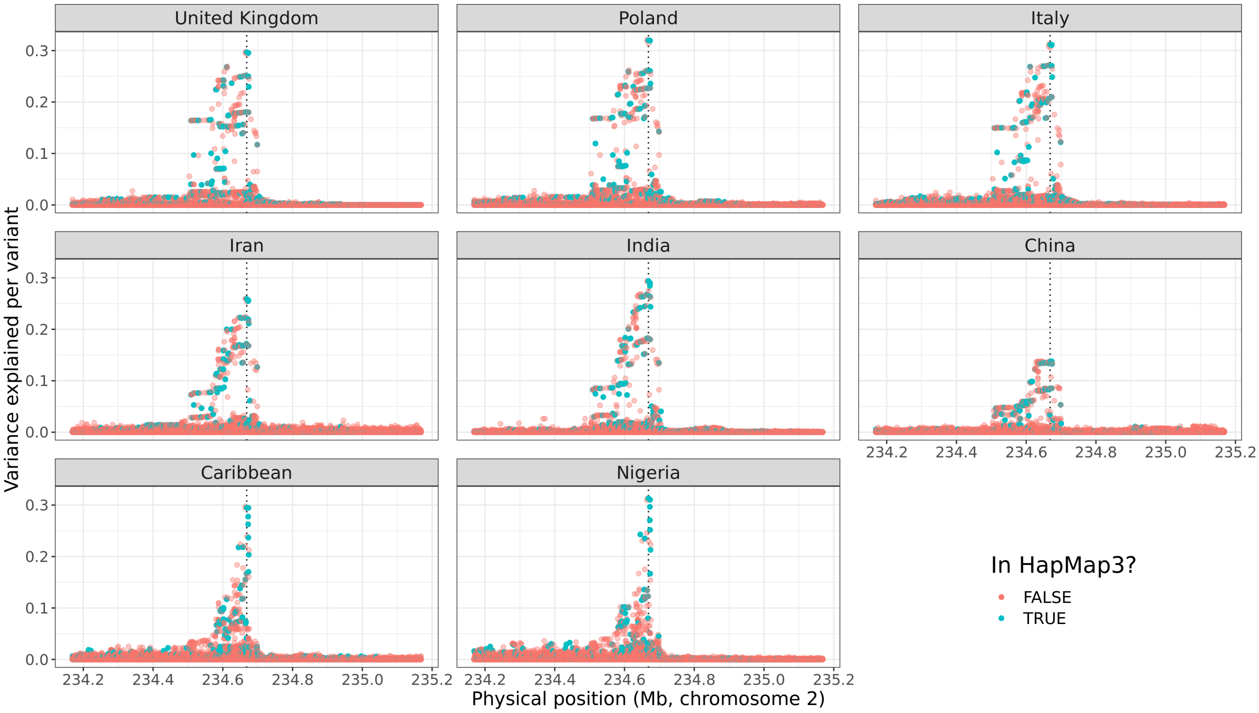


Figure S8: Zoomed Manhattan plot for total **bilirubin** concentration. The phenotypic variance explained per variant is computed as $r^2 = t^2/(n+t^2)$, where t is the t-score from GWAS and n is the degrees of freedom (the sample size minus the number of variables in the model, i.e. the covariates used in the GWAS, the intercept and the variant). The GWAS includes all variants with an imputation INFO score larger than 0.3 and within a 500Kb radius around the top hit from the GWAS performed in the UK training set and on the HapMap3 variants, represented by a vertical dotted line.

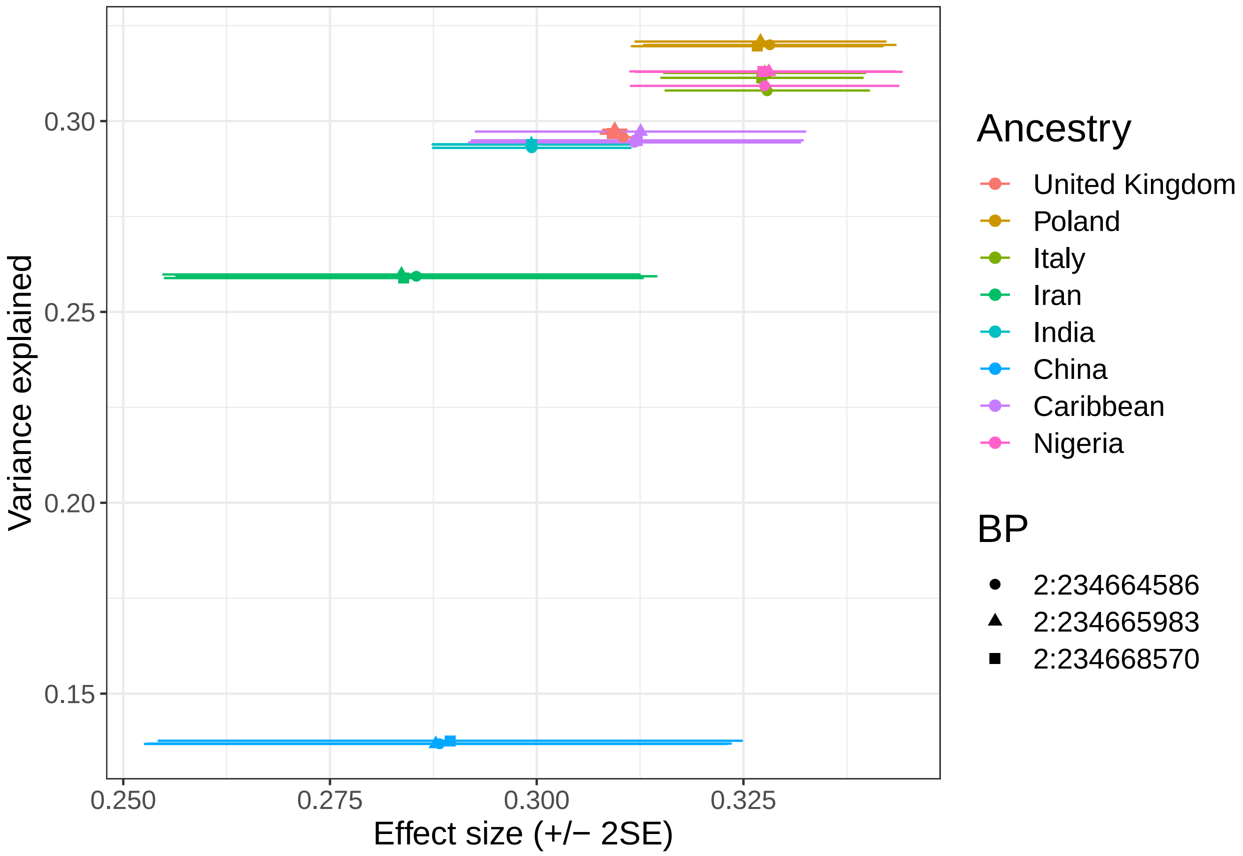


Figure S9: Effect sizes and variance explained for the top three variants from figure S8.

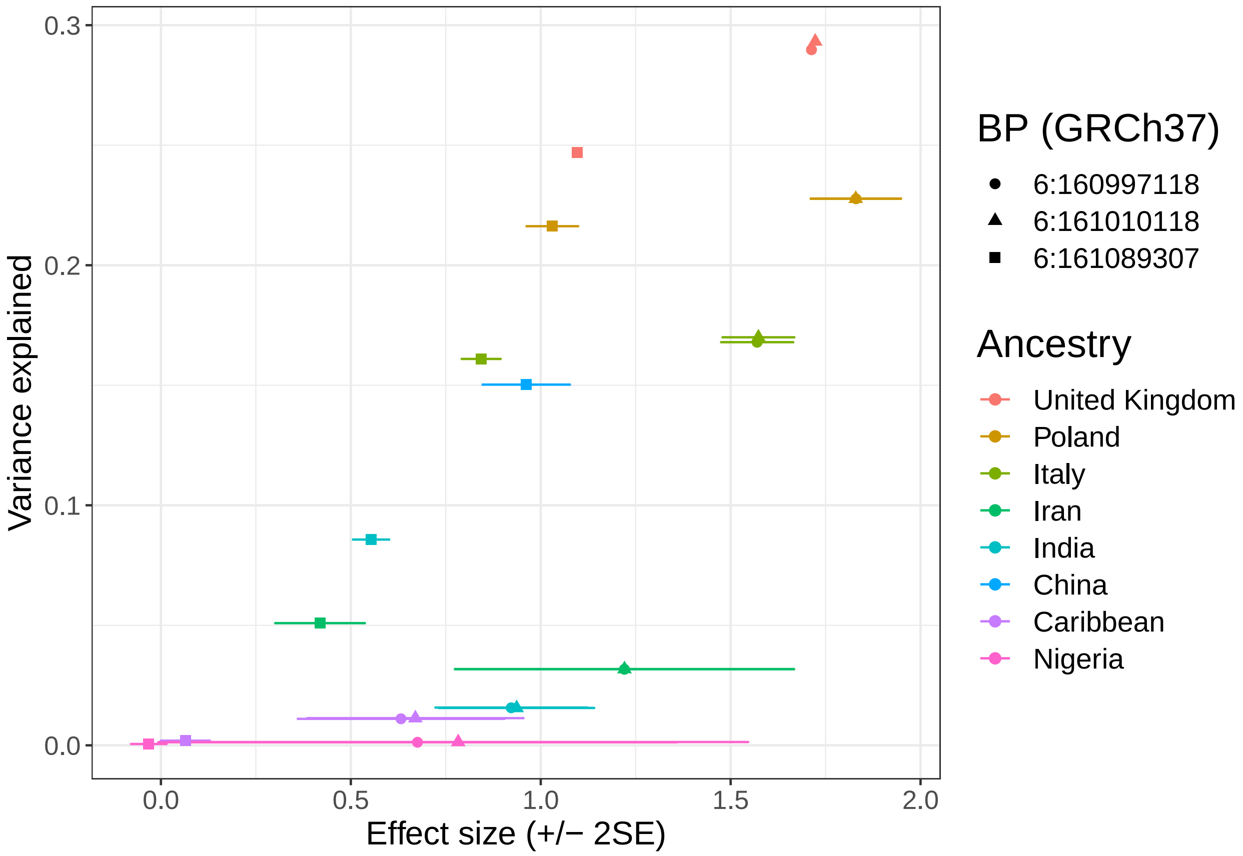


Figure S10: Effect sizes and variance explained for the top three variants from figure 4.

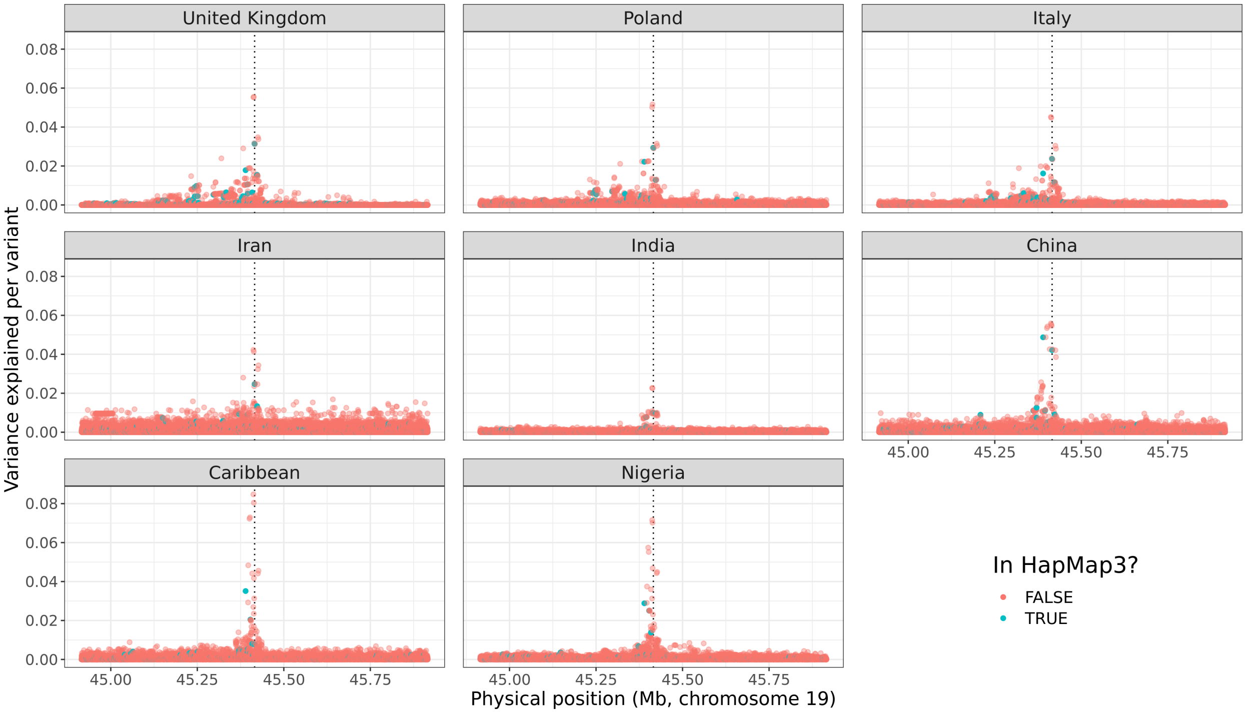


Figure S11: Zoomed Manhattan plot for **apolipoprotein B** concentration. The phenotypic variance explained per variant is computed as $r^2 = t^2/(n + t^2)$, where t is the t-score from GWAS and n is the degrees of freedom (the sample size minus the number of variables in the model, i.e. the covariates used in the GWAS, the intercept and the variant). The GWAS includes all variants with an imputation INFO score larger than 0.3 and within a 500Kb radius around the top hit from the GWAS performed in the UK training set and on the HapMap3 variants, represented by a vertical dotted line.

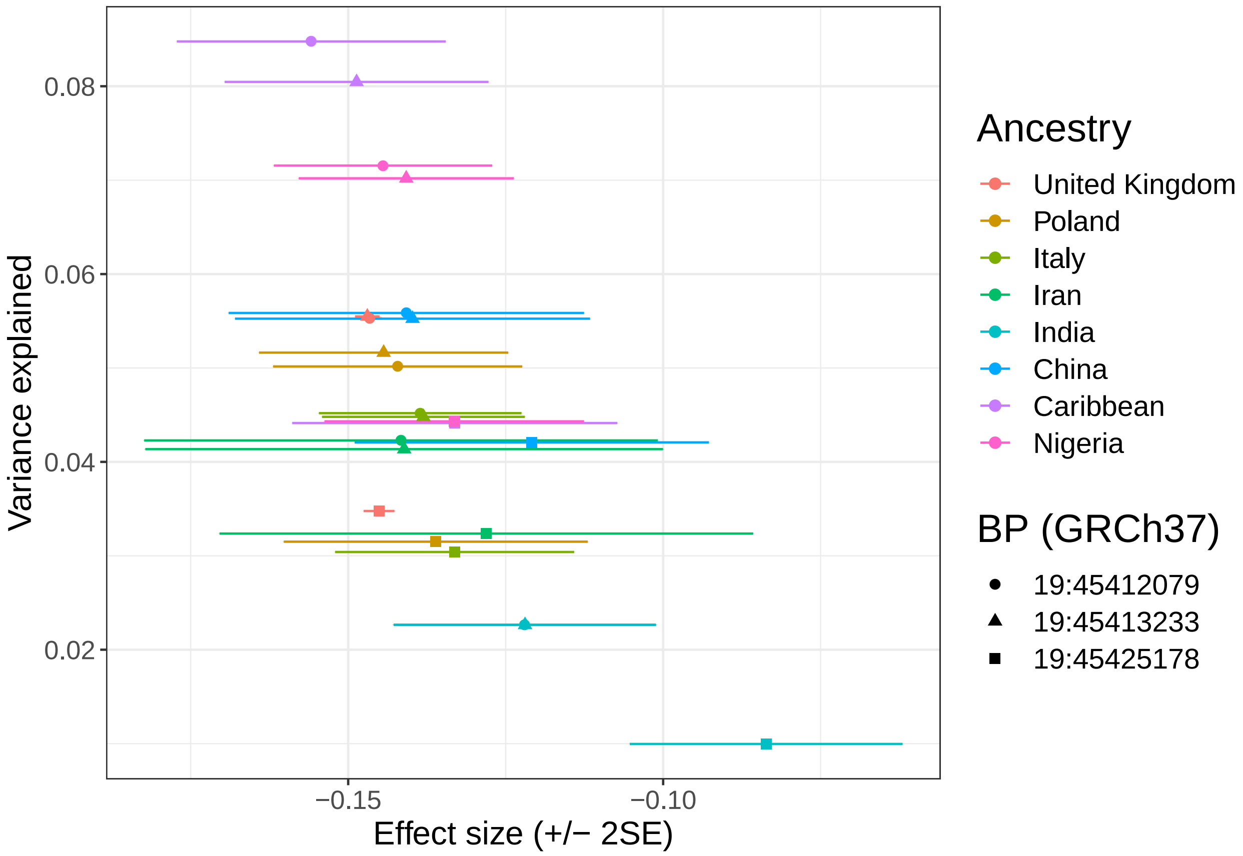


Figure S12: Effect sizes and variance explained for the top three variants from figure S11.

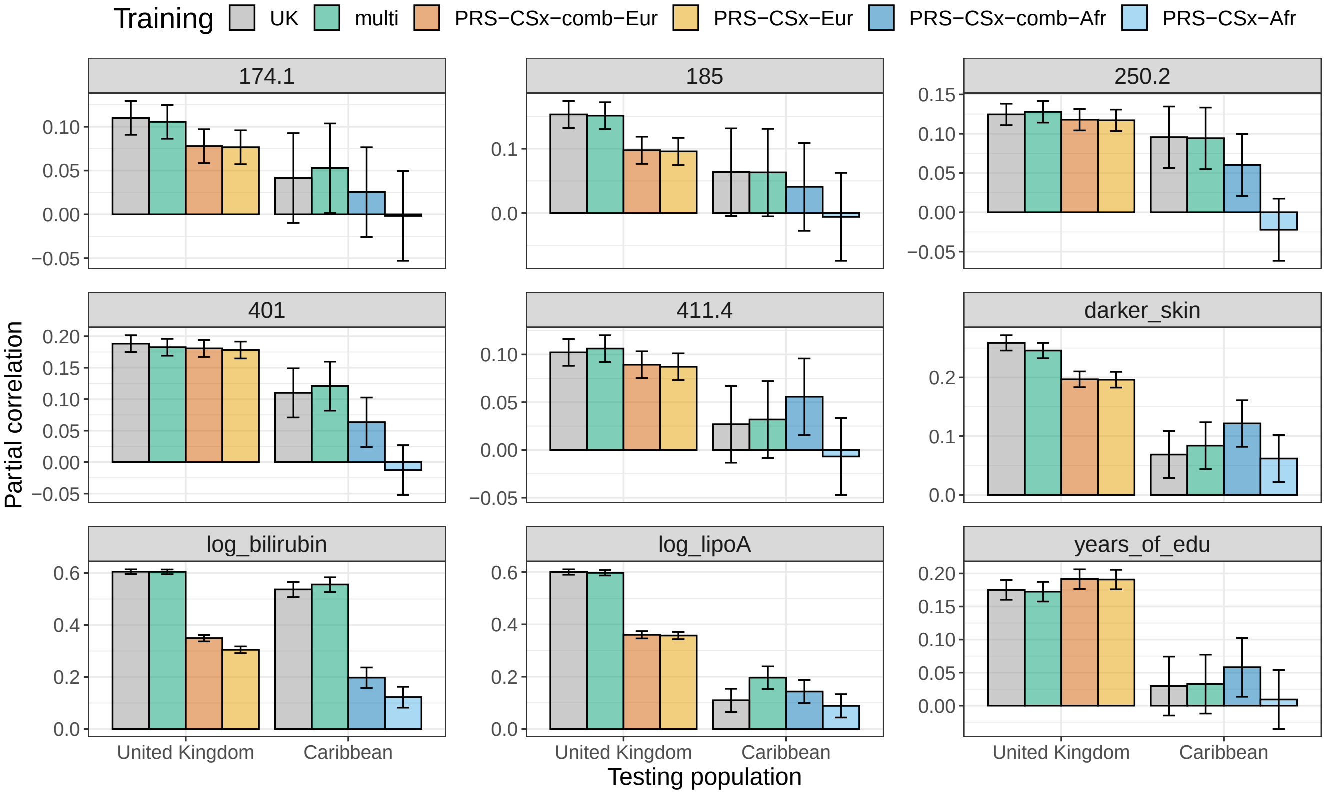


Figure S13: Partial correlation achieved per phenotype (each panel) and per ancestry group (x-axis) when training penalized regressions either with UK individuals only (training 1 in table 1) or when using individuals of multiple ancestries (training 2). We also run PRS-CSx on training 2, grouping the UK, Italy and Poland groups as a common European group, and removing the small Iran group as it does not have a similar ancestry as the four LD references provided for PRS-CSx. PRS-CSx-Eur corresponds to the raw prediction from PRS-CSx corresponding to the European ancestry data, while PRS-CSx-comb-Eur corresponds to the best prediction from the linear combination of the predictions for all four global ancestries on the UK test data. Phecode 174.1: breast cancer; 185: prostate cancer; 250.2: type 2 diabetes; 401: hypertension; 411.4: coronary artery disease.

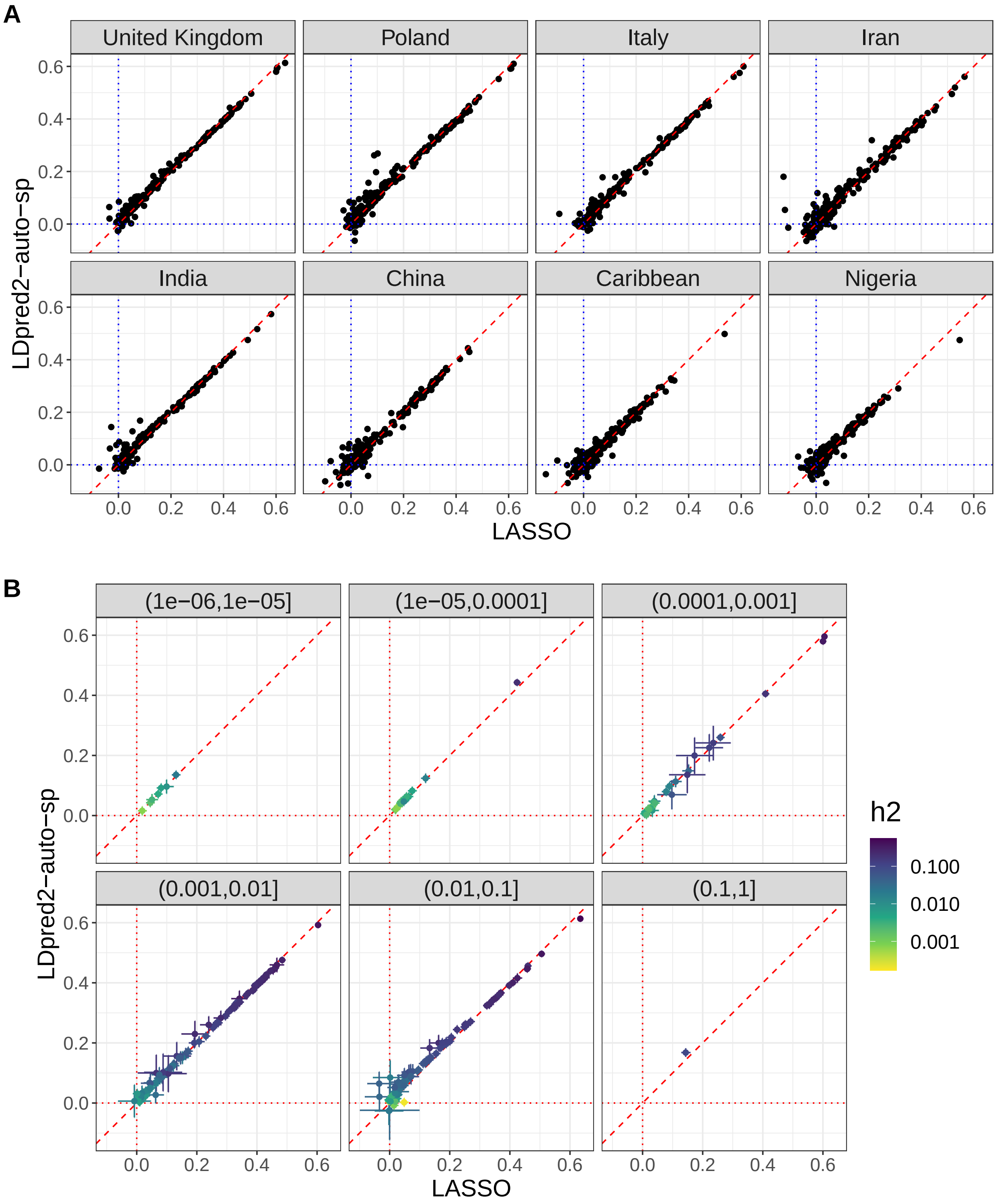


Figure S14: **A)** Partial correlations (and 95% CI) achieved per phenotype (each point) and per ancestry group (each panel) when training either with LASSO or with LDpred2-auto. **B)** Focusing now on the UK panel from A), each panel represents a range of proportion of causal variants p and points are colored by SNP heritability h^2 (estimates from LDpred2-auto). Penalized regression tends to provide better predictive performance than LDpred2 for phenotypes for which partial-$r > 0.3$, and inversely.

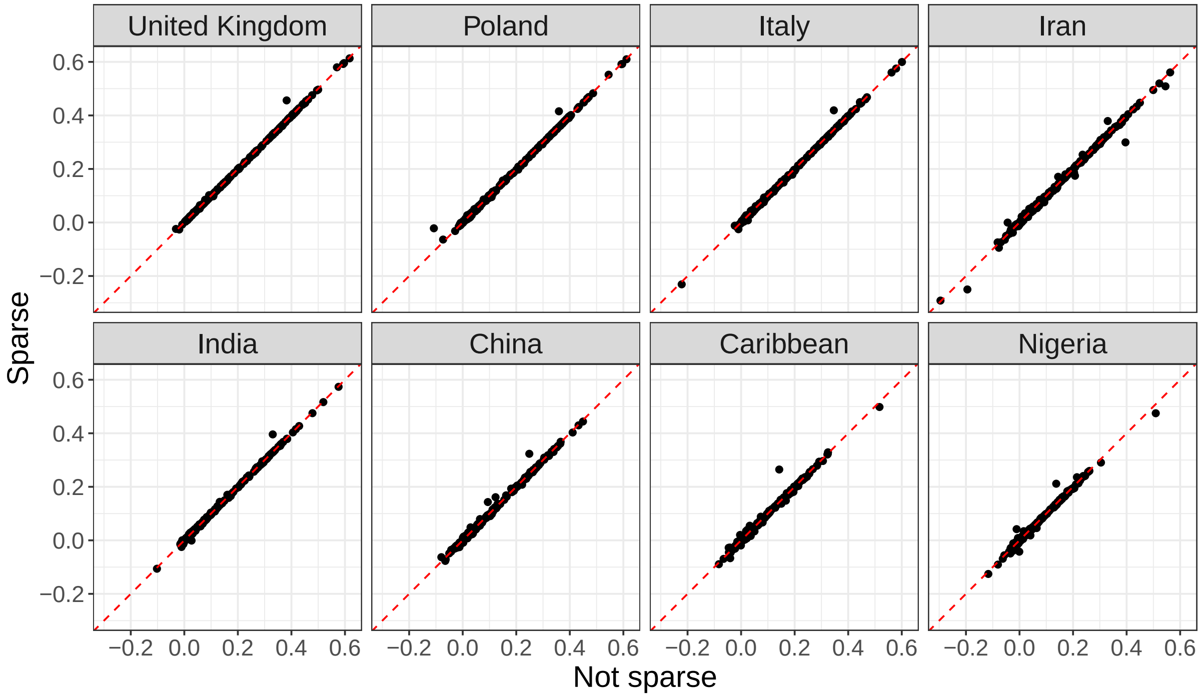


Figure S15: Partial correlations achieved per phenotype (each point) and per ancestry group (each panel) when training either with LDpred2-auto or with LDpred2-auto-sparse (sparse option enabled).

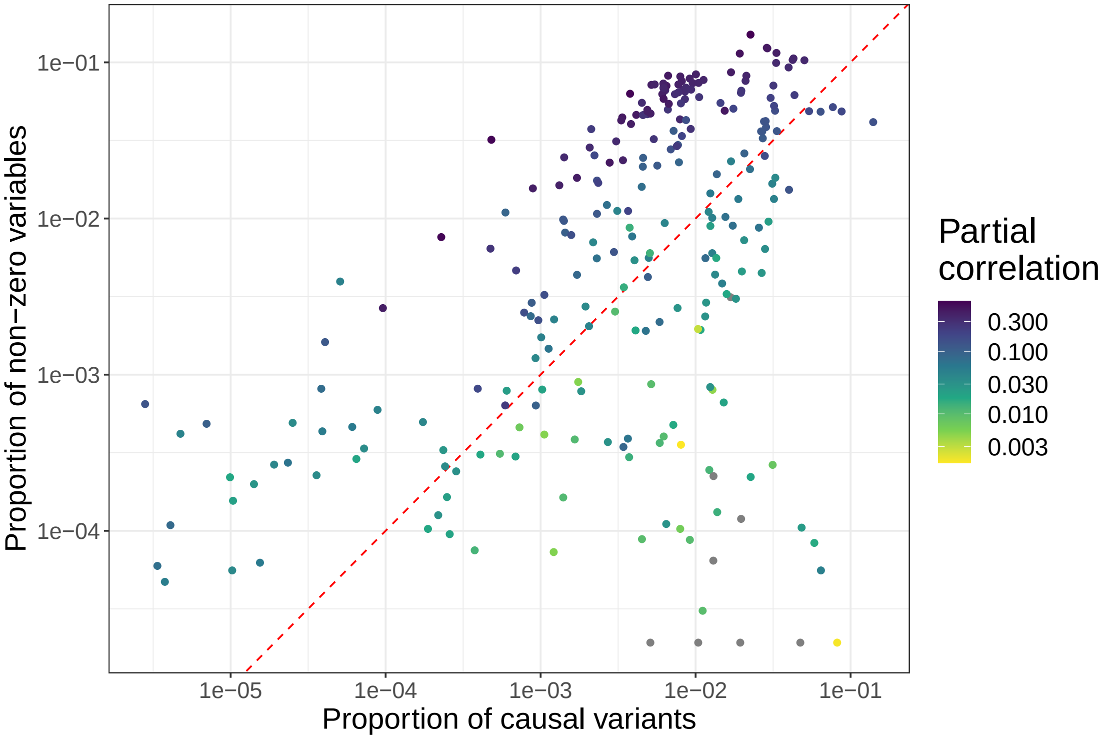


Figure S16: Proportion of variants with non-zero effects in the penalized regression models for each phenotype (point) versus the proportion of causal variants p estimated from LDpred2-auto, colored by the partial correlation achieved in the UK test set.

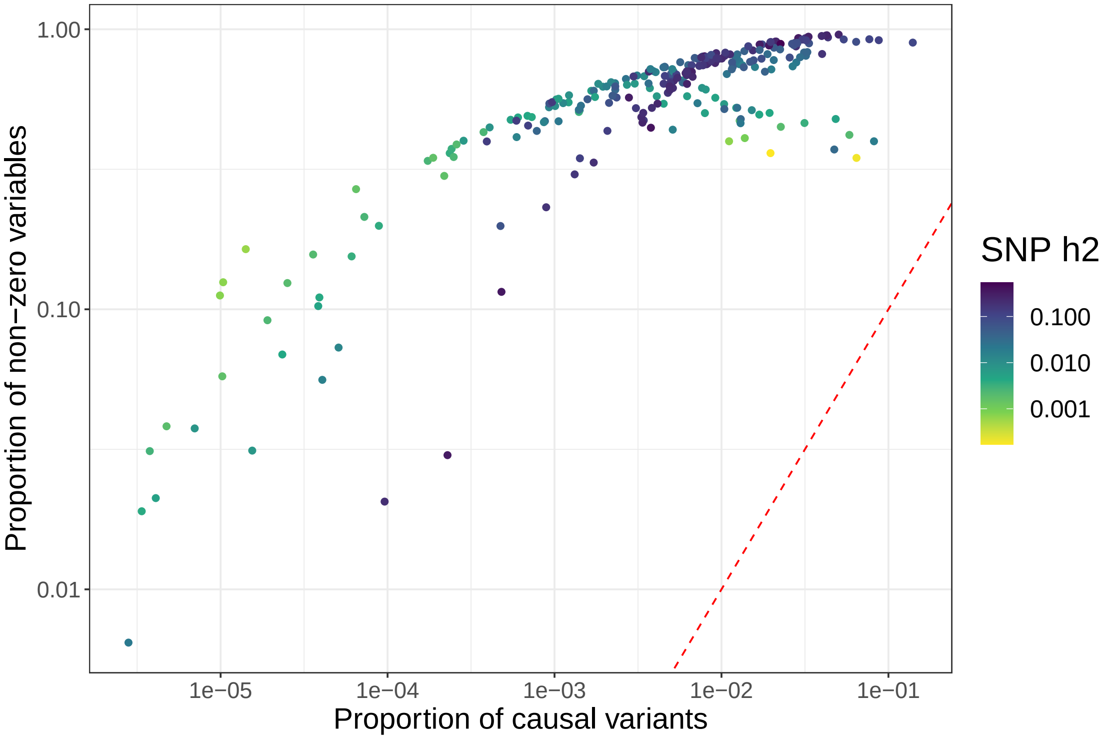


Figure S17: Proportion of variants with non-zero effects in LDpred2-auto-sparse for each phenotype (point) versus the proportion of causal variants p estimated from LDpred2-auto, colored by the SNP heritability h^2 estimated from LDpred2-auto.

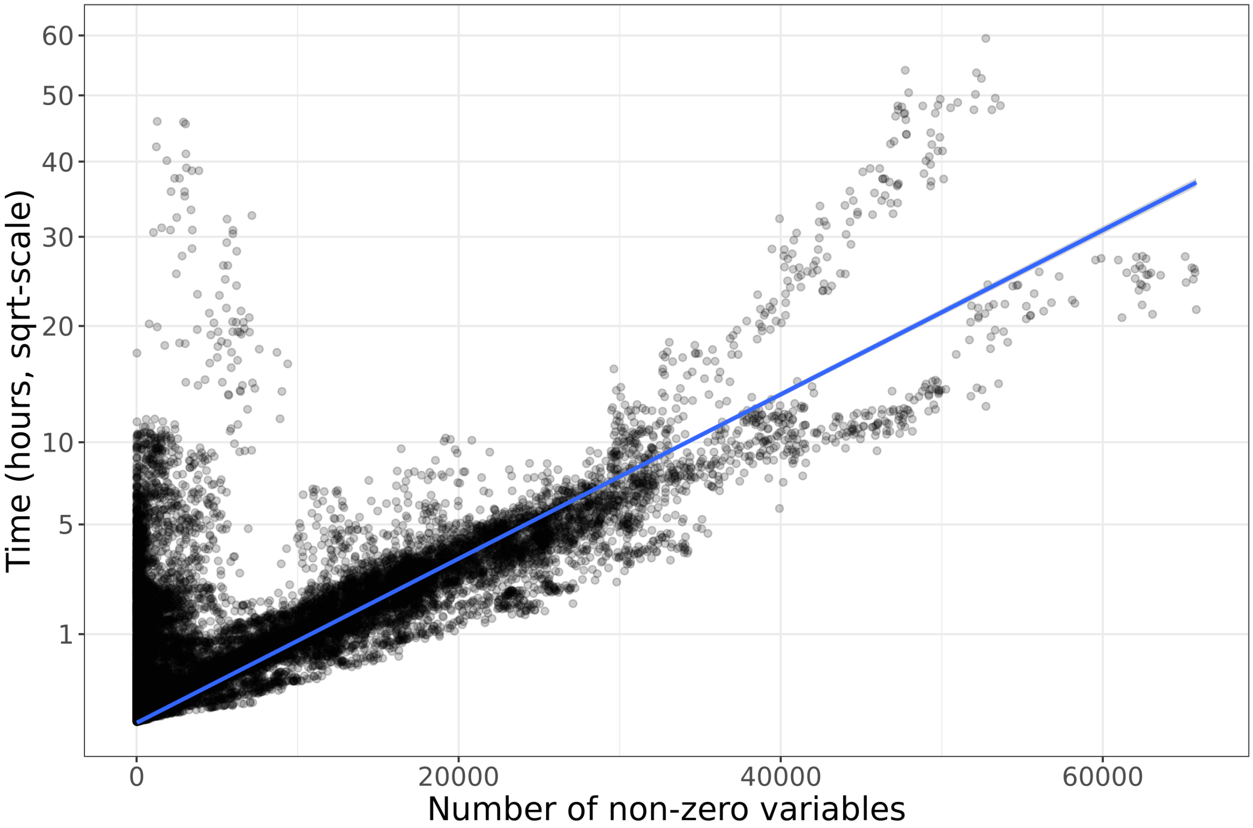


Figure S18: Computation times for all penalized regression models run using the 1M HapMap3 variants. We recall that we usually run 90 models for each phenotype because we use 9 sets of hyper-parameters and K=10 folds. Computation time is largely quadratic with the number of non-zero effects in the model. It is also dependent on the compute node and the loading of the HPC cluster at the time of running (Figure S19).

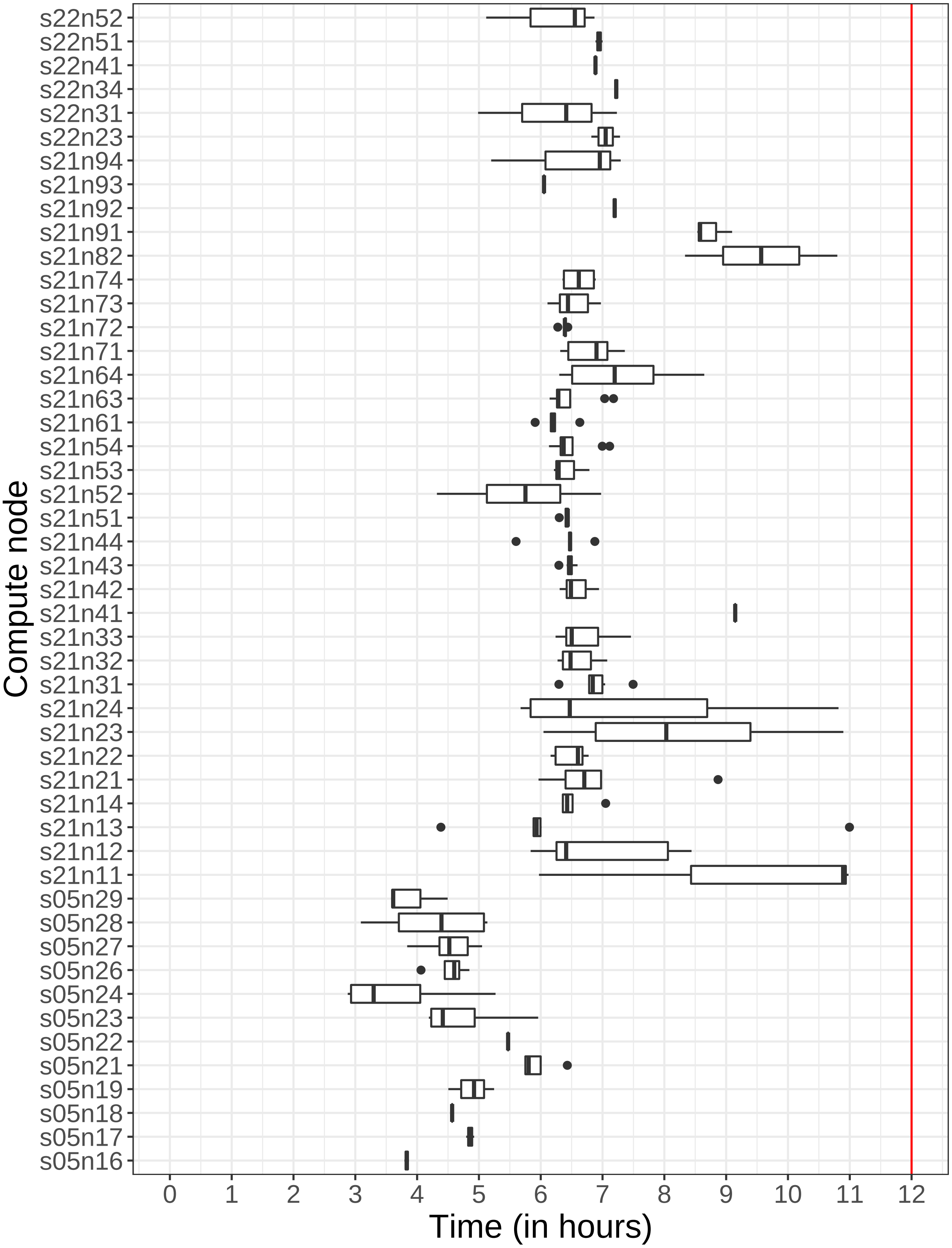


Figure S19: Computation times for fitting LDpred2-auto (with default 1000 burn-in iterations + 500 more + sparse option running 150 more) using the 1M HapMap3 variants. Running times should be the same for all phenotypes, yet we see some variability depending on the node used. Some fitting had to be run again because it exceeded the 12-hour timeout, which happened a few times when and the HPC cluster was particularly crowded.

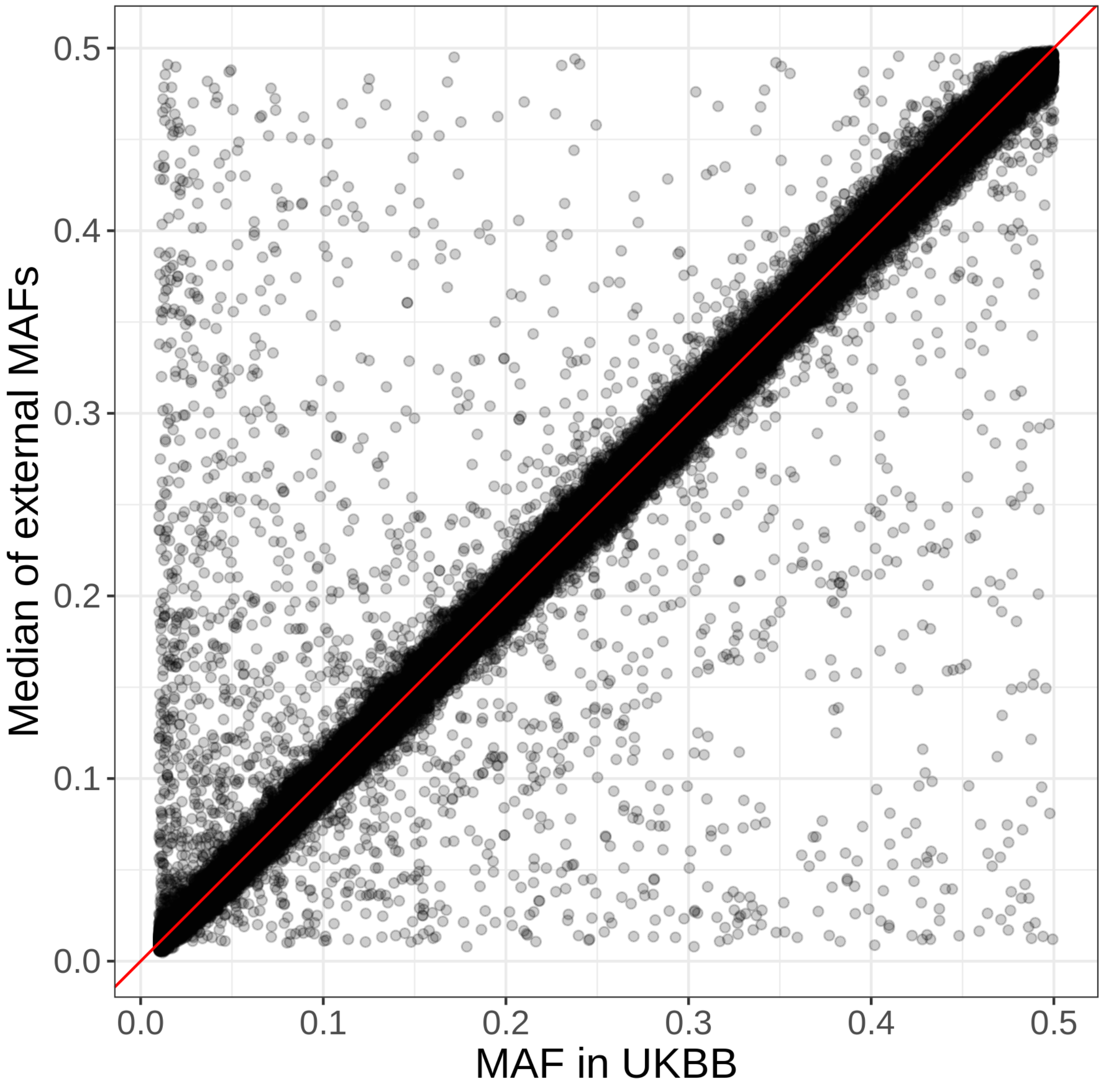


Figure S20: Comparison between frequencies in the UK Biobank and frequencies in external data.

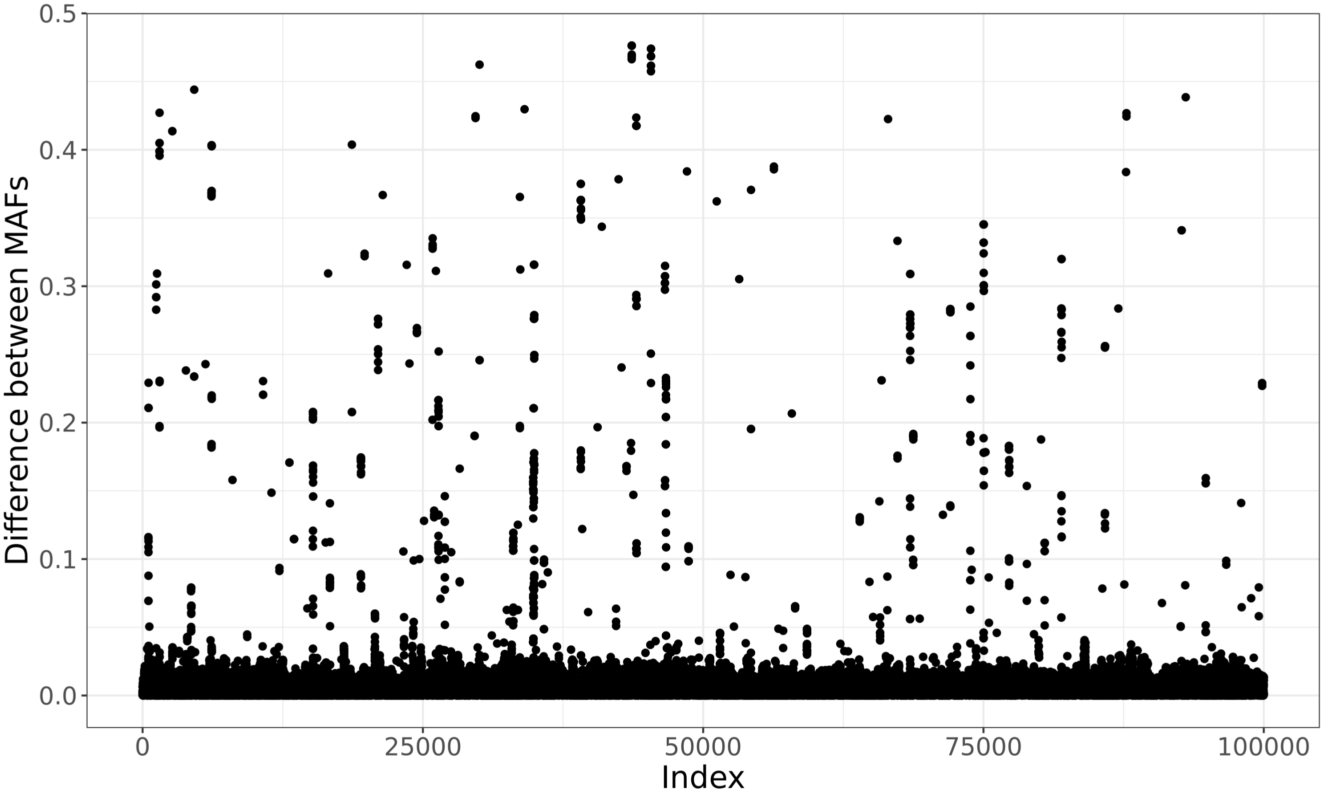


Figure S21: Differences in MAF between the first 100,000 variants in UK Biobank and external data. These differences (likely errors in UKBB) are hypothetically grouped around errors in the genotyped data that propagated to the imputed data.

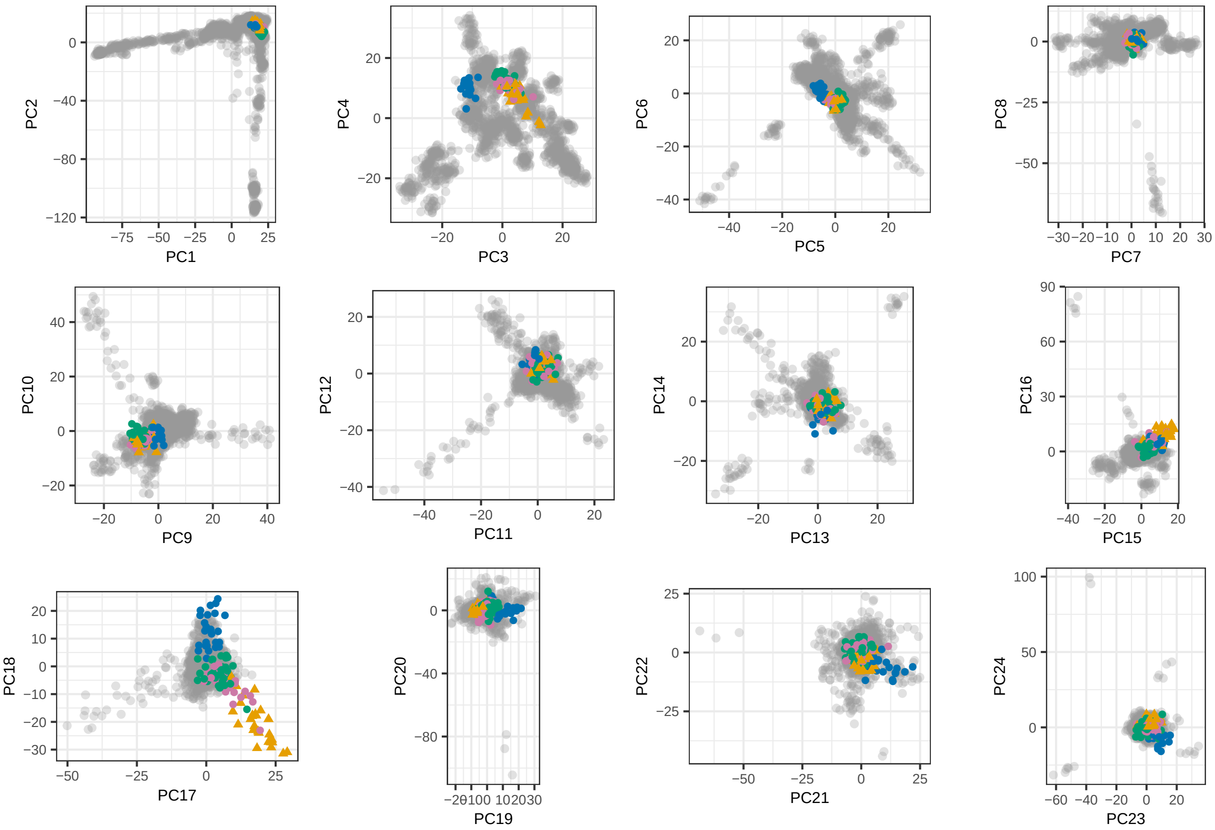


Figure S22: First 24 PC scores for the PCA computed in the reference dataset composed of several Jewish and non-Jewish individuals (Behar *et al.* 2013). Orange triangles represent the Ashkenazi Jews, pink points the Italian and Sephardi Jews, green points the Maghrebian Jews, and blue points the Iranian and Iraqi Jews.

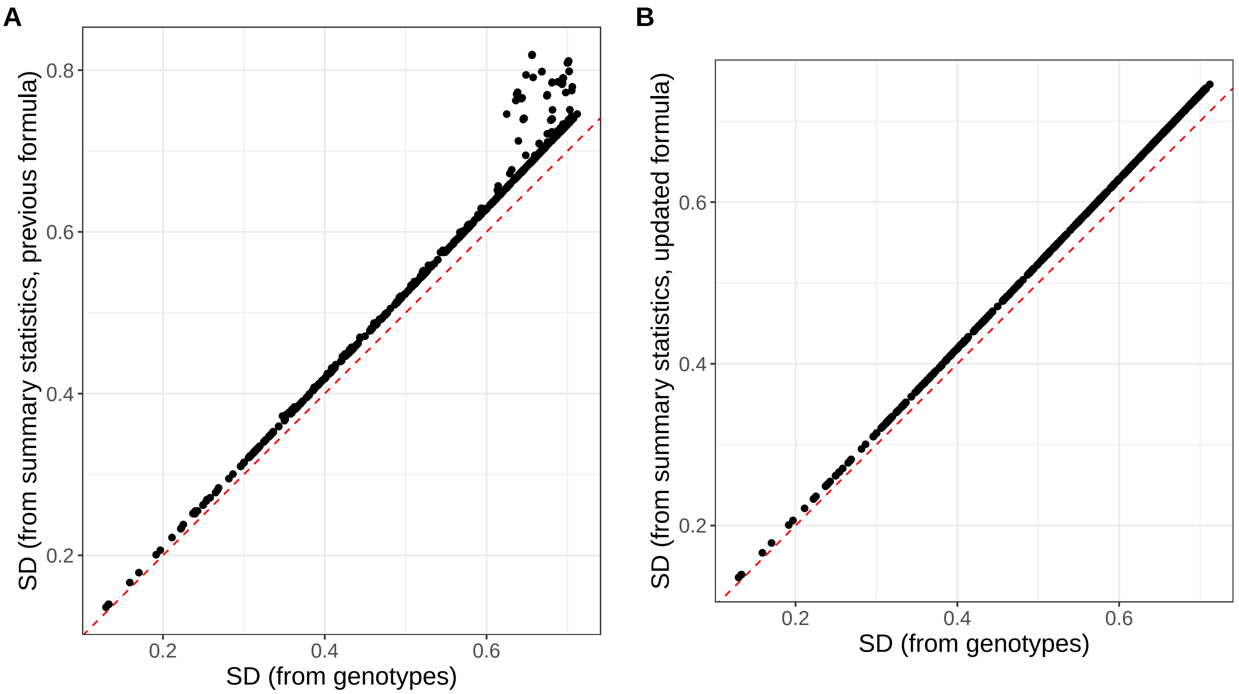


Figure S23: Comparison of the standard deviations (SD) computed from both genotypes and summary statistics for the 1000 most associated variants with bilirubin concentration. A) uses the previous formula $\text{sd}(\boldsymbol{G_j}) \approx \frac{\text{sd}(\boldsymbol{y})}{\sqrt{n \ \text{se}(\hat{\gamma}_j)^2}}$ proposed in Privé *et al.* (2020) while B) uses the updated formula $\text{sd}(\boldsymbol{G_j}) \approx \frac{\text{sd}(\boldsymbol{y})}{\sqrt{n \ \text{se}(\hat{\gamma}_j)^2 + \hat{\gamma}_j^2}}$ proposed here, which does one less approximation. The slope slightly larger than 1 can be explained by $\text{sd}(\boldsymbol{y}) > \text{sd}(\breve{\boldsymbol{y}})$.

Phenotype	Set of variants	h2 [2.5%-97.5%]	p [2.5%-97.5%]
174.1	top1M	0.0889 [0.086-0.092]	0.0076 [0.00678-0.00841]
174.1	HM3	0.0299 [0.0264-0.0334]	0.000881 [0.000636-0.00117]
185	top1M	0.113 [0.109-0.116]	0.00819 [0.00743-0.00906]
185	HM3	0.0381 [0.0343-0.0423]	0.000784 [0.000588-0.00105]
411.4	top1M	0.0641 [0.0624-0.0659]	0.0152 [0.0138-0.0168]
411.4	HM3	0.0401 [0.0379-0.0422]	0.00457 [0.00397-0.00526]
apoB	top1M	0.269 [0.265-0.272]	0.0533 [0.0498-0.0568]
apoB	HM3	0.163 [0.16-0.166]	0.00132 [0.00119-0.00145]
height	top1M	0.482 [0.479-0.486]	1 [1-1]
height	HM3	0.546 [0.541-0.552]	0.0226 [0.0218-0.0235]
log_bilirubin	top1M	0.301 [0.267-0.363]	0.214 [0.195-0.227]
log_bilirubin	HM3	0.361 [0.357-0.365]	0.000481 [0.000423-0.000545]
log_BMI	top1M	0.173 [0.171-0.176]	1 [1-1]
log_BMI	HM3	0.263 [0.26-0.266]	0.0426 [0.0404-0.0446]
log_lipoA	top1M	0.696 [0.689-0.702]	0.0116 [0.011-0.0122]
log_lipoA	HM3	0.34 [0.336-0.345]	0.000229 [0.000192-0.000268]

Table S1: Estimates of SNP heritability h^2 and proportion of causal variants p from LDpred2-auto, when using either 1,040,096 HapMap3 variants or when prioritizing 1M variants out of 8M+ common variants, for eight phenotypes. Quantiles of all estimates are also reported. Phecode 174.1: breast cancer; 185: prostate cancer; 411.4: coronary artery disease.

References

Alexander, D. H., Novembre, J., and Lange, K. (2009). Fast model-based estimation of ancestry in unrelated individuals. *Genome Research*, **19**(9), 1655–1664.

Behar, D. M., Metspalu, M., Baran, Y., Kopelman, N. M., Yunusbayev, B., Gladstein, A., Tzur, S., Sahakyan, H., Bahmanimehr, A., Yepiskoposyan, L., *et al.* (2013). No evidence from genome-wide data of a Khazar origin for the Ashkenazi Jews. *Human Biology*, **85**(6), 859–900.

Privé, F., Arbel, J., and Vilhjálmsson, B. J. (2020). LDpred2: better, faster, stronger. *Bioinformatics*, **36**(22-23), 5424–5431.

Supplementary Note: Ancestry inference and grouping from principal component analysis of genetic data

[†] Further defined in section "Definitions † and methods".

Measures of genetic dissimilarity between populations

We first compare four measures of genetic dissimilarity using populations of the 1000 Genomes Project (1000G[†], 1000 Genomes Project Consortium *et al.* (2015)). The F_{ST}[†] is an ubiquitous measure of genetic dissimilarity between populations and the first measure we use in this comparison. We report F_{ST} between the 26 1000G populations in tables SA2-SA6, and the clustering of these populations based on F_{ST} in figure SA3. The other three measures compared are distances applied to the PC scores[†] of the genetic data: 1) the Bhattacharyya distance[†]; 2) the distance between the centers (geometric medians[†]) of the two populations; and 3) the shortest distance between pairs of PC scores, one from each of the two populations. The (squared) Euclidean distance between population centers appears to be an appropriate PCA-based distance as it is approximately proportional to the F_{ST} (Figure SA1) and provides an appropriate clustering of populations (Figure SA6). However, future work is needed to understand why residuals are bimodal for large distances (e.g. in figure SA1). This relation between F_{ST} and (squared) Euclidean distances in the PCA space has been previously shown for two populations only (McVean 2009).

Previously, we and others proposed to use (robust) Mahalanobis distances to infer ancestry or identify a single homogeneous group of individuals (Peterson *et al.* 2017; Privé *et al.* 2020). When looking at distances between two populations, this corresponds to using the Bhattacharyya distance. However, in contrast to Euclidean distances, the two other Bhattacharyya and shortest distances do not provide as satisfactory results (Figures SA4, SA5, SA8 and SA9). For example, African Caribbeans in Barbados (ACB) and Americans of African Ancestry in SW USA (ASW) and the four admixed American (AMR) populations are close to all European (EUR), South Asian (SAS) and African (AFR) populations when using the Bhattacharyya distance (Figure SA4). We hypothesize that the main issue with this approach is that an admixed population covers a large volume in the PCA space, therefore all distances to this population cluster are small because of the

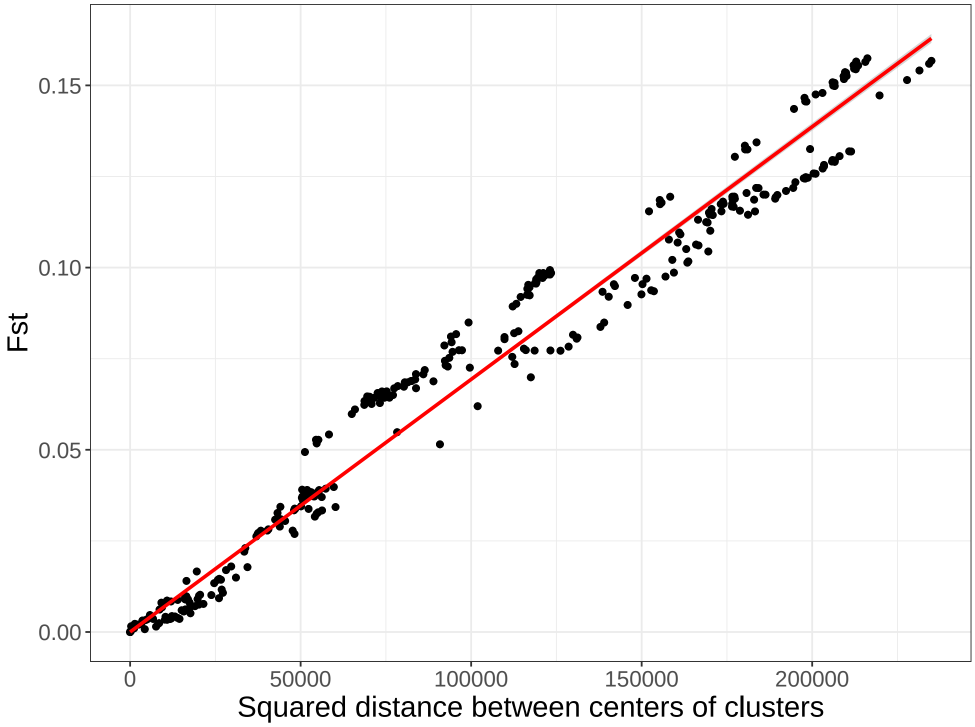


Figure SA1: Comparing F_{ST} to the squared Euclidean distance on the PCA space (i.e. using PC scores†) between centers of pairs of the 26 1000G populations.

covariance component from the Mahalanobis distance. In contrast, the global scale of the PC scores used when using Euclidean distances is invariant from the cluster scattering.

We also vary the number of PCs used for computing the Euclidean distances and how they compare with F_{ST} in figure SA7. With 2 to 4 PCs, we are able to adequately separate distant populations, but not the closest ones. For example, when using 4 PCs, there are pairs of populations with an F_{ST} of ~0.02 while their PC centers are superimposed (Figure SA7). When using more PCs (8, 16 or 25) to compute the distances, results remain mostly similar.

PCA-based ancestry inference

We project the dataset of interest onto the PCA space of the 1000G data using the fast tools developed in Privé *et al.* (2020). We recall that this uses an automatic removal of LD when computing PCA and a correction for shrinkage in projected PC scores, which has been shown to be particularly important when using PC scores for ancestry estimation (Zhang *et al.* 2020). Based on the results from the previous section, we propose to assign individual ancestry to one of the 26 1000G populations based on the Euclidean distance to these reference population centers in the PCA space (geometric medians† of PC scores†). Since we showed previously that (squared) distances in the PCA space are proportional to F_{ST}, we can set a threshold on these distances that would correspond approximately to an F_{ST} of e.g. 0.002. This threshold is close to the dissimilarity between Spanish and Italian people (F_{ST}(IBS, TSI) of 0.0015). When an individual is not close enough to any of the

26 1000G populations, we leave its ancestry inference as unknown, otherwise we assign this individual to the closest reference population center.

We first perform ancestry estimation for the individuals in the UK Biobank[†]. For 488,371 individuals, this procedure takes less than 20 minutes using 16 cores. These individuals seem to originate from many parts of the world when we project onto the PCA space of the 1000G (Figure SA10). Self-reported ancestry (Field 21000) is available for almost all individuals, with only 1.6% with unknown or mixed ancestry. When using the threshold defined before, we could not infer ancestry for 4.6% of all 488,371 individuals. More precisely, among "British", "Irish" and "White" ancestries, this represented respectively 2.2%, 3.3% and 7.9% (Tables SA7 and SA9). This also represented 3.3% for "Chinese", 13.8% for "Indian" and 17.8% for "African" ancestries. Finally, mixed ancestries were particularly difficult to match to any of the 1000G populations, e.g. 97.3% unmatched within "White and Black Africa" and 93.0% within "White and Asian" ancestries. Only 47 individuals were misclassified in "super" population of the 1000G; e.g. six "British" were classified as South Asians, one "Chinese" as European and 25 "Caribbean" as South Asian by our method (Table SA7). However, when comparing the location of these mismatched individuals to the rest of individuals on the PCA space computed within the UK Biobank (Bycroft *et al.* 2018), it seems more probable that our genetic ancestry estimate is exact while the self-reported ancestry is not matching the underlying genetic ancestry for these individuals (Figure SA11). This possible discrepancy between self-reported ancestry and genetic ancestry has been reported before (Mersha and Abebe 2015).

We also test the approach proposed in Zhang *et al.* (2020) which consists in finding the 20 nearest neighbors in 1000G and computing the frequency of (super) population membership, weighted by the inverse distance to these 20 closest 1000G individuals. When this probability is less than 0.875, they leave the ancestry as unknown, aiming at discarding admixed individuals. Less than 0.5% could not be matched by their method (Table SA8). Of note, they could match much more admixed individuals, whereas they set a high probability threshold aiming at discarding such admixed individuals. Morever, there are many more discrepancies between their method and the self-reported ancestry in the UK Biobank (Table SA8) compared to the previous results with our method (Table SA7). The global scale used in Euclidean distances makes it more robust to infer ancestry as compared to using relative proportions from k=20 nearest neighbors (kNN, Zhang *et al.* (2020)). Indeed, consider e.g. an admixed individual of say 25% European ancestry and 75% African ancestry. The kNN-based method is likely to identify this individual as of African ancestry, while our method will probably be unable to match it, which is a beneficial feature when we are interested in defining genetically homogeneous groups. We also believe our proposed method to be more robust than machine learning methods, because a machine learning method would try e.g. to differentiate between GBR and CEU 1000G populations, which are two very close populations of Northwest Europe (F_{ST} of 0.0002). In other words, our distance-based method should benefit from the inclusion of any new reference population, whereas it would make it increasingly complex to apply machine learning methods.

Finally, our method is able to accurately differentiate between sub-continental populations such as differentiating between Pakistani, Bangladeshi and Chinese people (Table SA9). We also applied our ancestry detection technique to the European individuals of the POPRES data (Nelson *et al.* 2008). Only 16 out of the 1385 individuals (1.2%) could not be matched, of which 11 were from East or South-East Europe (Table SA10). Note that all individuals that we could match were identified as of European ancestry. We could also identify accurately sub-regions of Europe; e.g. 261 out of 264 Spanish and Portugese individuals were identified as "Iberian Population in Spain" (EUR_IBS, Table SA10).

The proposed method has two possible limitations. First, since we match target individuals to 1000G populations, if individuals are far from all 26 1000G populations, then they would not be matched. When looking at the POPRES data, more individuals from East Europe could not be matched. This is not surprising because there are no East European population in the 1000G data. Moreover, if we look at the location of the 1000G populations on a map, we can see that it lacks representation of many parts of the world (Figure SA12). This issue has also been reported e.g. for Asian populations (Lu and Xu 2013). Therefore more diverse populations should be aggregated to better cover the worldwide genome diversity, such as with the Simons Genome Diversity Project (Mallick *et al.* 2016), which would also improve the proposed method. A second potential limitation of the proposed method is that it has two hyper-parameters: the number of PCs used to compute the distances and the threshold on the minimum distance to any cluster center above which the ancestry is not matched. Several studies have used only the first two PCs for ancestry inference. We have shown here that using two PCs (or even four) is not enough for distinguishing between populations at the sub-continental level (Figure SA7). As in Privé *et al.* (2020), we recommend to use all PCs that visually separate some populations. Moreover, we believe our proposed method to be robust to increasing the number of PCs used because contribution to the Euclidean distance is smaller for later PCs than for first PCs. As for the distance limit, we have shown here how to define it to approximately correspond to an F_{ST} of 0.002. Alternatively, a threshold can be chosen based on the visual inspection of the histogram of distances (on a log scale). This threshold can also be adjusted depending on how homogeneous one want each cluster to be.

PCA-based ancestry grouping

Finally, we show several ways how to use our ancestry inference method for grouping genetically homogeneous individuals. One first possible approach is to simply match individuals that are close enough to one of the 1000G populations, as described previously. Alternatively, one could use the internal PC scores and the self-reported ancestries or countries of birth, e.g. available in the UK Biobank (Fields 21000 and 20115). **This solution does not require projecting individuals to the 1000G, but does require computing PC scores within the dataset instead.** In the UK Biobank data, we can define centers of the seven self-reported ancestry groups: British, Indian, Pakistani, Bangladeshi, Chinese, Caribbean and African; then match all individuals to one of these centers (or none if an individual is far from all centers). This enables e.g. to capture a larger set of individuals who are close enough to British people (e.g. Irish people), while discarding individuals whose genetic ancestry is not matching the self-reported ancestry (Table SA11). Only 3.7% of all individuals could not be matched. The resulting clusters are presented in the PCA space in figure SA13.

One could do the same using the countries of birth instead of the self-reported ancestries, which we use in the main text. Again, the country of birth may sometime not reflect the ancestral origin. Therefore, we first compute the robust centers (geometric medians) of all countries with at least 300 individuals. Then, we cluster these countries based on their distance in the PCA space to make sure of their validity as proxies for genetic ancestry and to choose a small subset of centers with good coverage of the overall dissimilarities (Figure SA14). Based on the previous clustering and the available sample sizes, we choose to use the centers from the following eight countries as reference: the United Kingdom, Poland, Iran, Italy, India, China, "Caribbean" and Nigeria. Only 2.8% of all individuals could not be matched to one of these eight groups (Table SA1). The resulting clusters are presented in the PCA space in figure SA2. Note that these clusters probably include individuals from nearby countries as well. Moreover, more clusters could probably be defined, e.g. the individuals with large values for PC6 in figure SA2 seem to originate from South America with many people from Colombia, Chile, Mexico, Peru, Ecuador, Venezuela, Bolivia, Brazil, and Argentina. However, here we decide to restrict to large enough clusters (e.g. with more than 1000 individuals). The cluster with small values for PC4 corresponds to Ashkenazi ancestry, and is described in the main text.

Finally, when we know that the data is composed of a predominant ancestry, we can define a single homogeneous cluster by simply restricting to individuals who are close enough to the overall center of all individuals (Figure SA15). When doing so, we can cluster 91% of the data into one cluster composed of 421,871 British, 12,039 Irish, 8351 "Other White", 1814 individuals of unknown ancestry, 467 "White" and 41 individuals of other self-reported ancestries. This is made possible because we use the geometric median which is robust to outliers.

Table SA1: Self-reported ancestry (left) of UKBB individuals and their matching to country groups (top) by our method.

	United Kingdom	Poland	Iran	Italy	India	China	Caribbean	Nigeria	Not matched
British	423509	1412	30	3152	18	1	2		2890
Irish	12683	14		29					27
White	472	13	8	38		1			13
Other White	8102	2754	239	3259	2				1459
Indian	6		33		4296				1381
Pakistani	1		2		1672				73
Bangladeshi					4				217
Chinese	1					1441			62
Other Asian	4	1	226	3	299	93		1	1120
Caribbean					3		2306	1245	743
African	1				2		71	2281	849
Other Black					2		36	34	46
Asian or Asian British			4		23	2			13
Black or Black British	2						11	9	4
White and Black Caribbean	7			3			13	1	573
White and Black African	6			4			1	2	389
White and Asian	56		12	30	54				650
Unknown	1827	116	680	462	345	315	215	513	3347

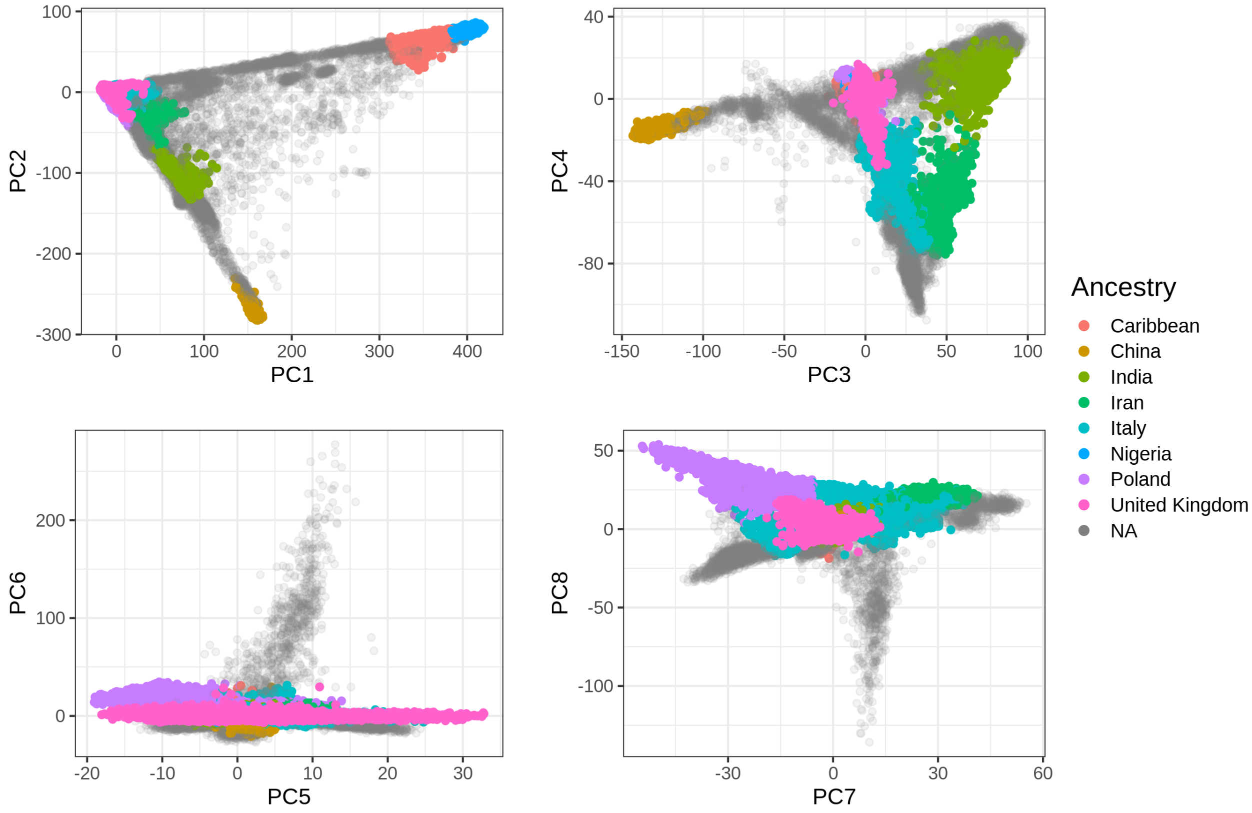


Figure SA2: The first eight PC scores† computed from the UK Biobank (Field 22009) colored by the homogeneous ancestry group we infer for these individuals.

Definitions † and methods

Note that the code used in this supplementary note is available at `https://github.com/privefl/paper-ancestry-matching/tree/master/code`.

- The **1000 Genomes Project (1000G)** data is composed of approximately 100 individuals for each of 26 populations worldwide (described at `https://www.internationalgenome.org/category/population/`), including 7 African (AFR), 5 East Asian (EAS), 5 South Asian (SAS), 5 European (EUR) and 4 admixed American (AMR) populations. Here we used the transformed data in PLINK format provided in Privé *et al.* (2020).

- The $\boldsymbol{F_{ST}}$ measures the relative amount of genetic variance between populations compared to the total genetic variance within these populations (Wright 1965). We use the weighted average formula proposed in Weir and Cockerham (1984), which we now implement in our package bigsnpr (Privé *et al.* 2018).

- The **Principal Component (PC) scores** are defined as $U\Delta$, where $U\Delta V^T$ is the singular value decomposition of the (scaled) genotype matrix (Privé *et al.* 2020). They are usually truncated, e.g. corresponding to the first 20 principal dimensions only.

- The **Bhattacharyya distance** between two multivariate normal distributions $\mathcal{N}(\boldsymbol{\mu}_1,\, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2,\, \boldsymbol{\Sigma}_2)$ is defined as $D_B = \frac{1}{8}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \frac{1}{2}\log\left(\frac{|\boldsymbol{\Sigma}|}{\sqrt{|\boldsymbol{\Sigma}_1|\,|\boldsymbol{\Sigma}_2|}}\right)$, where $\boldsymbol{\Sigma} = \frac{\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2}{2}$ and $|M|$ is the absolute value of the determinant of matrix M (Bhattacharyya 1943; Fukunaga 1990). The mean and covariance parameters for each population are computed using the robust location and covariance parameters as proposed in Privé *et al.* (2020).

- The **geometric median** of points is the point that minimizes the sum of all Euclidean distances to these points. We now implement this as function `geometric_median` in our R package bigutilsr.

- The **UK Biobank** is a large cohort of half a million individuals from the UK, for which we have access to both genotypes and multiple phenotypes (`https://www.ukbiobank.ac.uk/`). We apply some quality control filters to the genotyped data; we remove individuals with more than 10% missing values, variants with more than 1% missing values, variants having a minor allele frequency < 0.01, variants with P-value of the Hardy-Weinberg exact test < 10^{-50}, and non-autosomal variants. This results in 488,371 individuals and 504,139 genetic variants.

References

1000 Genomes Project Consortium *et al.* (2015). A global reference for human genetic variation. *Nature*, **526**(7571), 68.

Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, **35**, 99–109.

Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L. T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., *et al.* (2018). The UK Biobank resource with deep phenotyping and genomic data. *Nature*, **562**(7726), 203.

Fukunaga, K. (1990). Introduction to statistical pattern recognition, ser. *Computer science and scientific computing. Boston: Academic Press*.

Lu, D. and Xu, S. (2013). Principal component analysis reveals the 1000 Genomes Project does not sufficiently cover the human genetic diversity in Asia. *Frontiers in genetics*, **4**, 127.

Mallick, S., Li, H., Lipson, M., Mathieson, I., Gymrek, M., Racimo, F., Zhao, M., Chennagiri, N., Nordenfelt, S., Tandon, A., *et al.* (2016). The simons genome diversity project: 300 genomes from 142 diverse populations. *Nature*, **538**(7624), 201–206.

McVean, G. (2009). A genealogical interpretation of principal components analysis. *PLoS Genet*, **5**(10), e1000686.

Mersha, T. B. and Abebe, T. (2015). Self-reported race/ethnicity in the age of genomic research: its potential impact on understanding health disparities. *Human genomics*, **9**(1), 1.

Nelson, M. R., Bryc, K., King, K. S., Indap, A., Boyko, A. R., Novembre, J., Briley, L. P., Maruyama, Y., Waterworth, D. M., Waeber, G., *et al.* (2008). The population reference sample, POPRES: a resource for population, disease, and pharmacological genetics research. *The American Journal of Human Genetics*, **83**(3), 347–358.

Peterson, R. E., Edwards, A. C., Bacanu, S.-A., Dick, D. M., Kendler, K. S., and Webb, B. T. (2017). The utility of empirically assigning ancestry groups in cross-population genetic studies of addiction. *The American journal on addictions*, **26**(5), 494–501.

Privé, F., Aschard, H., Ziyatdinov, A., and Blum, M. G. B. (2018). Efficient analysis of large-scale genome-wide data with two R packages: bigstatsr and bigsnpr. *Bioinformatics*, **34**(16), 2781–2787.

Privé, F., Luu, K., Blum, M. G. B., McGrath, J. J., and Vilhjálmsson, B. J. (2020). Efficient toolkit implementing best practices for principal component analysis of population genetic data. *Bioinformatics*. btaa520.

Weir, B. S. and Cockerham, C. C. (1984). Estimating F-statistics for the analysis of population structure. *Evolution*, pages 1358–1370.

Wright, S. (1965). The interpretation of population structure by F-statistics with special regard to systems of mating. *Evolution*, pages 395–420.

Zhang, D., Dey, R., and Lee, S. (2020). Fast and robust ancestry prediction using principal component analysis. *Bioinformatics*, **36**(11), 3439–3446.

Additional Figures and Tables

Measures of genetic dissimilarity between populations

Table SA2: F_{ST} values between African populations of the 1000G and all 26 1000G populations.

	LWK	ESN	YRI	ACB	ASW	GWD	MSL
LWK	0.0000	0.0077	0.0071	0.0064	0.0090	0.0108	0.0093
ESN	0.0077	0.0000	0.0008	0.0034	0.0088	0.0075	0.0051
YRI	0.0071	0.0008	0.0000	0.0025	0.0080	0.0062	0.0039
ACB	0.0064	0.0034	0.0025	0.0000	0.0020	0.0060	0.0044
ASW	0.0090	0.0088	0.0080	0.0020	0.0000	0.0098	0.0094
GWD	0.0108	0.0075	0.0062	0.0060	0.0098	0.0000	0.0036
MSL	0.0093	0.0051	0.0039	0.0044	0.0094	0.0036	0.0000
JPT	0.1475	0.1564	0.1545	0.1344	0.1194	0.1517	0.1574
CHB	0.1456	0.1546	0.1527	0.1324	0.1174	0.1499	0.1556
CHS	0.1466	0.1555	0.1536	0.1335	0.1186	0.1509	0.1565
CDX	0.1456	0.1544	0.1526	0.1324	0.1178	0.1498	0.1555
KHV	0.1435	0.1525	0.1507	0.1304	0.1154	0.1479	0.1535
GIH	0.1101	0.1200	0.1186	0.0954	0.0773	0.1156	0.1200
PJL	0.1069	0.1167	0.1154	0.0920	0.0735	0.1124	0.1167
BEB	0.1077	0.1174	0.1161	0.0934	0.0755	0.1131	0.1174
ITU	0.1096	0.1195	0.1181	0.0954	0.0778	0.1151	0.1195
STU	0.1091	0.1189	0.1175	0.0949	0.0774	0.1145	0.1189
PEL	0.1472	0.1559	0.1541	0.1325	0.1144	0.1515	0.1567
MXL	0.1125	0.1219	0.1205	0.0972	0.0772	0.1175	0.1218
CLM	0.0970	0.1063	0.1051	0.0816	0.0620	0.1021	0.1061
PUR	0.0849	0.0938	0.0927	0.0699	0.0515	0.0898	0.0935
FIN	0.1219	0.1319	0.1306	0.1044	0.0837	0.1272	0.1319
CEU	0.1189	0.1291	0.1278	0.1014	0.0805	0.1244	0.1290
GBR	0.1193	0.1295	0.1282	0.1017	0.0808	0.1248	0.1294
IBS	0.1145	0.1247	0.1234	0.0975	0.0772	0.1199	0.1247
TSI	0.1154	0.1258	0.1245	0.0986	0.0783	0.1210	0.1258

Table SA3: F_{ST} values between admixed American populations of the 1000G and all 26 1000G populations.

	PEL	MXL	CLM	PUR
LWK	0.1472	0.1125	0.0970	0.0849
ESN	0.1559	0.1219	0.1063	0.0938
YRI	0.1541	0.1205	0.1051	0.0927
ACB	0.1325	0.0972	0.0816	0.0699
ASW	0.1144	0.0772	0.0620	0.0515
GWD	0.1515	0.1175	0.1021	0.0898
MSL	0.1567	0.1218	0.1061	0.0935
JPT	0.0795	0.0643	0.0707	0.0773
CHB	0.0786	0.0628	0.0689	0.0752
CHS	0.0811	0.0650	0.0708	0.0769
CDX	0.0849	0.0675	0.0719	0.0773
KHV	0.0817	0.0643	0.0689	0.0744
GIH	0.0725	0.0370	0.0278	0.0269
PJL	0.0688	0.0327	0.0230	0.0220
BEB	0.0669	0.0344	0.0278	0.0282
ITU	0.0732	0.0391	0.0308	0.0303
STU	0.0728	0.0390	0.0309	0.0305
PEL	0.0000	0.0170	0.0380	0.0548
MXL	0.0170	0.0000	0.0090	0.0180
CLM	0.0380	0.0090	0.0000	0.0056
PUR	0.0548	0.0180	0.0056	0.0000
FIN	0.0772	0.0338	0.0178	0.0149
CEU	0.0804	0.0334	0.0143	0.0100
GBR	0.0809	0.0338	0.0146	0.0102
IBS	0.0820	0.0339	0.0134	0.0081
TSI	0.0825	0.0345	0.0143	0.0090

Table SA4: F_{ST} values between East Asian populations of the 1000G and all 26 1000G populations.

	JPT	CHB	CHS	CDX	KHV
LWK	0.1475	0.1456	0.1466	0.1456	0.1435
ESN	0.1564	0.1546	0.1555	0.1544	0.1525
YRI	0.1545	0.1527	0.1536	0.1526	0.1507
ACB	0.1344	0.1324	0.1335	0.1324	0.1304
ASW	0.1194	0.1174	0.1186	0.1178	0.1154
GWD	0.1517	0.1499	0.1509	0.1498	0.1479
MSL	0.1574	0.1556	0.1565	0.1555	0.1535
JPT	0.0000	0.0068	0.0086	0.0166	0.0140
CHB	0.0068	0.0000	0.0010	0.0084	0.0062
CHS	0.0086	0.0010	0.0000	0.0047	0.0031
CDX	0.0166	0.0084	0.0047	0.0000	0.0016
KHV	0.0140	0.0062	0.0031	0.0016	0.0000
GIH	0.0693	0.0673	0.0685	0.0685	0.0650
PJL	0.0669	0.0647	0.0660	0.0660	0.0626
BEB	0.0542	0.0518	0.0528	0.0527	0.0494
ITU	0.0656	0.0636	0.0647	0.0646	0.0611
STU	0.0642	0.0623	0.0634	0.0633	0.0598
PEL	0.0795	0.0786	0.0811	0.0849	0.0817
MXL	0.0643	0.0628	0.0650	0.0675	0.0643
CLM	0.0707	0.0689	0.0708	0.0719	0.0689
PUR	0.0773	0.0752	0.0769	0.0773	0.0744
FIN	0.0924	0.0901	0.0920	0.0925	0.0893
CEU	0.0985	0.0960	0.0977	0.0978	0.0946
GBR	0.0993	0.0968	0.0985	0.0985	0.0953
IBS	0.0981	0.0957	0.0973	0.0973	0.0942
TSI	0.0981	0.0956	0.0972	0.0972	0.0940

Table SA5: F_{ST} values between European populations of the 1000G and all 26 1000G populations.

	FIN	CEU	GBR	IBS	TSI
LWK	0.1219	0.1189	0.1193	0.1145	0.1154
ESN	0.1319	0.1291	0.1295	0.1247	0.1258
YRI	0.1306	0.1278	0.1282	0.1234	0.1245
ACB	0.1044	0.1014	0.1017	0.0975	0.0986
ASW	0.0837	0.0805	0.0808	0.0772	0.0783
GWD	0.1272	0.1244	0.1248	0.1199	0.1210
MSL	0.1319	0.1290	0.1294	0.1247	0.1258
JPT	0.0924	0.0985	0.0993	0.0981	0.0981
CHB	0.0901	0.0960	0.0968	0.0957	0.0956
CHS	0.0920	0.0977	0.0985	0.0973	0.0972
CDX	0.0925	0.0978	0.0985	0.0973	0.0972
KHV	0.0893	0.0946	0.0953	0.0942	0.0940
GIH	0.0343	0.0325	0.0328	0.0334	0.0317
PJL	0.0289	0.0269	0.0272	0.0278	0.0262
BEB	0.0372	0.0368	0.0372	0.0375	0.0362
ITU	0.0393	0.0380	0.0384	0.0384	0.0367
STU	0.0398	0.0385	0.0389	0.0389	0.0373
PEL	0.0772	0.0804	0.0809	0.0820	0.0825
MXL	0.0338	0.0334	0.0338	0.0339	0.0345
CLM	0.0178	0.0143	0.0146	0.0134	0.0143
PUR	0.0149	0.0100	0.0102	0.0081	0.0090
FIN	0.0000	0.0062	0.0066	0.0101	0.0116
CEU	0.0062	0.0000	0.0002	0.0022	0.0034
GBR	0.0066	0.0002	0.0000	0.0024	0.0037
IBS	0.0101	0.0022	0.0024	0.0000	0.0015
TSI	0.0116	0.0034	0.0037	0.0015	0.0000

Table SA6: F_{ST} values between South Asian populations of the 1000G and all 26 1000G populations.

	GIH	PJL	BEB	ITU	STU
LWK	0.1101	0.1069	0.1077	0.1096	0.1091
ESN	0.1200	0.1167	0.1174	0.1195	0.1189
YRI	0.1186	0.1154	0.1161	0.1181	0.1175
ACB	0.0954	0.0920	0.0934	0.0954	0.0949
ASW	0.0773	0.0735	0.0755	0.0778	0.0774
GWD	0.1156	0.1124	0.1131	0.1151	0.1145
MSL	0.1200	0.1167	0.1174	0.1195	0.1189
JPT	0.0693	0.0669	0.0542	0.0656	0.0642
CHB	0.0673	0.0647	0.0518	0.0636	0.0623
CHS	0.0685	0.0660	0.0528	0.0647	0.0634
CDX	0.0685	0.0660	0.0527	0.0646	0.0633
KHV	0.0650	0.0626	0.0494	0.0611	0.0598
GIH	0.0000	0.0035	0.0042	0.0039	0.0043
PJL	0.0035	0.0000	0.0035	0.0033	0.0036
BEB	0.0042	0.0035	0.0000	0.0022	0.0021
ITU	0.0039	0.0033	0.0022	0.0000	0.0013
STU	0.0043	0.0036	0.0021	0.0013	0.0000
PEL	0.0725	0.0688	0.0669	0.0732	0.0728
MXL	0.0370	0.0327	0.0344	0.0391	0.0390
CLM	0.0278	0.0230	0.0278	0.0308	0.0309
PUR	0.0269	0.0220	0.0282	0.0303	0.0305
FIN	0.0343	0.0289	0.0372	0.0393	0.0398
CEU	0.0325	0.0269	0.0368	0.0380	0.0385
GBR	0.0328	0.0272	0.0372	0.0384	0.0389
IBS	0.0334	0.0278	0.0375	0.0384	0.0389
TSI	0.0317	0.0262	0.0362	0.0367	0.0373

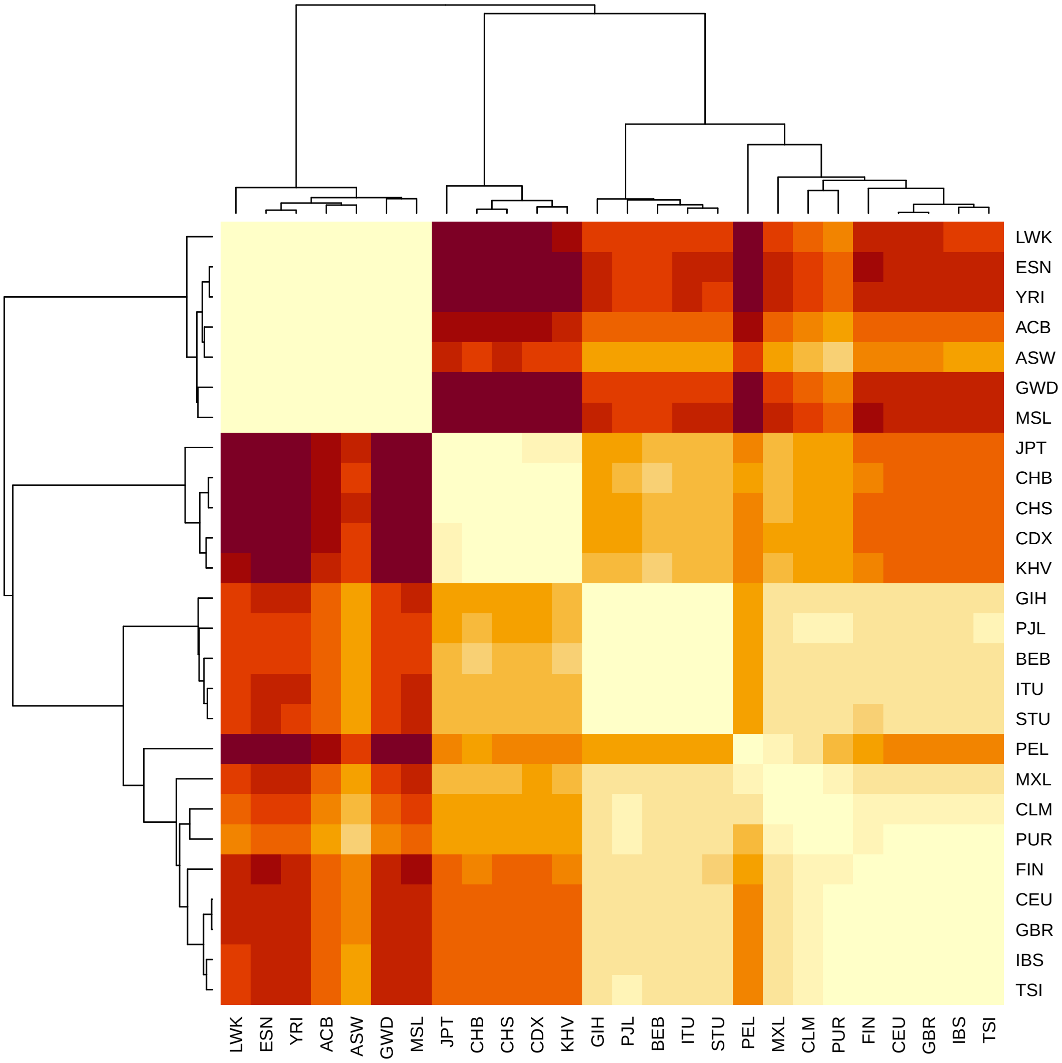


Figure SA3: Heatmap with clustering based on the F_{ST} between pairs of the 26 1000G populations. Corresponding values are reported in tables SA2-SA6.

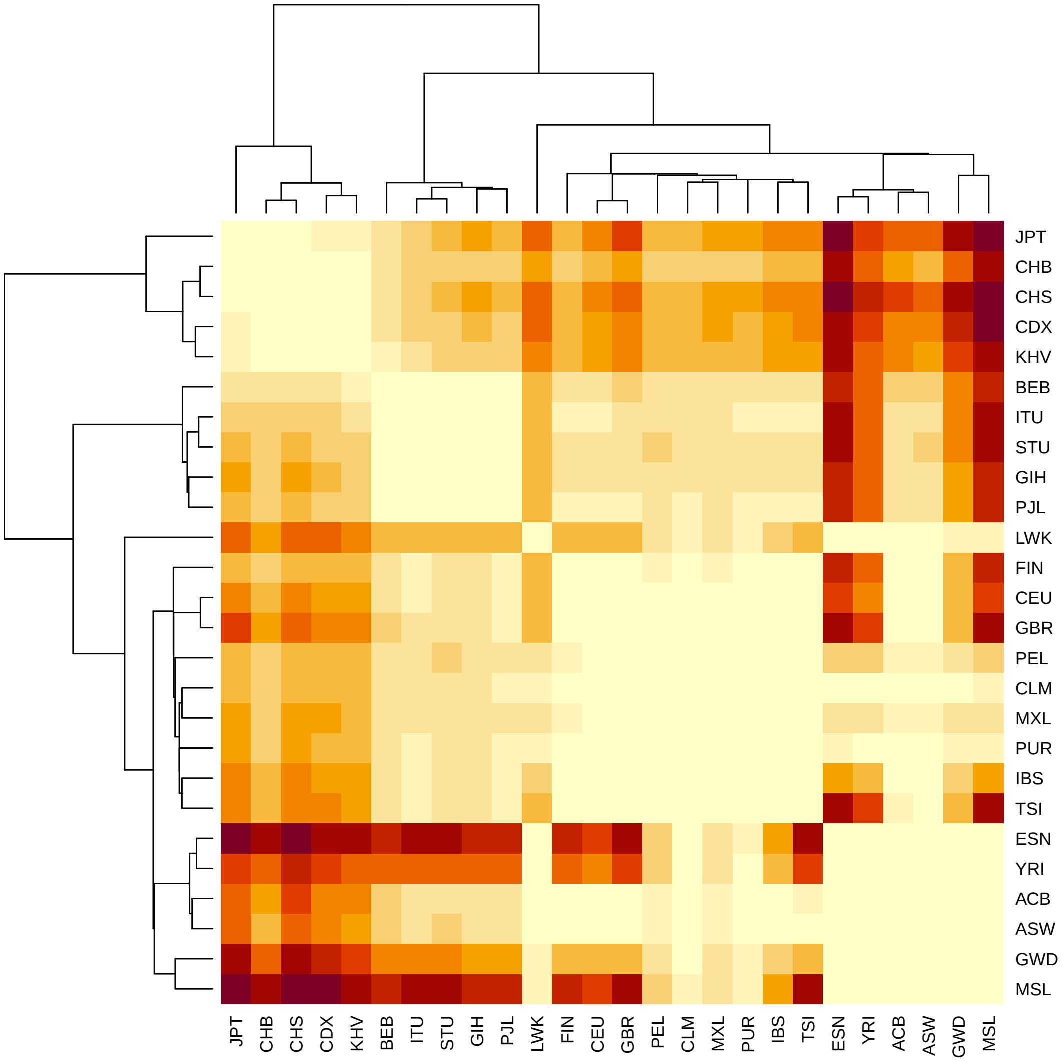


Figure SA4: Heatmap with clustering based on the Bhattacharyya distances between pairs of the 26 1000G populations.

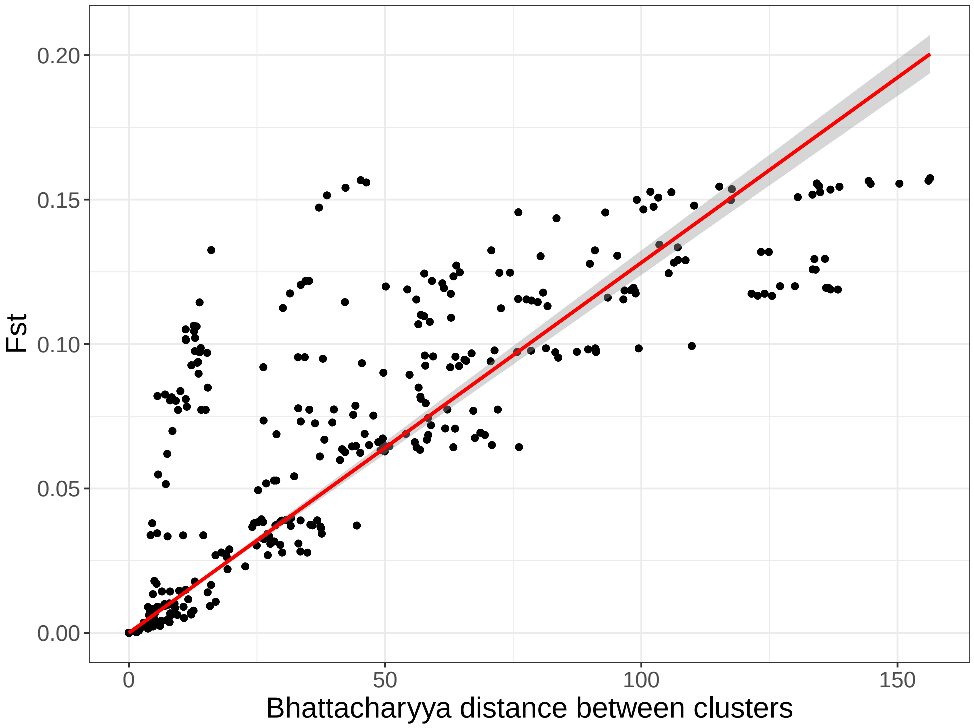


Figure SA5: Comparing F_{ST} to the Bhattacharyya distance on the PCA space between pairs of the 26 1000G populations.

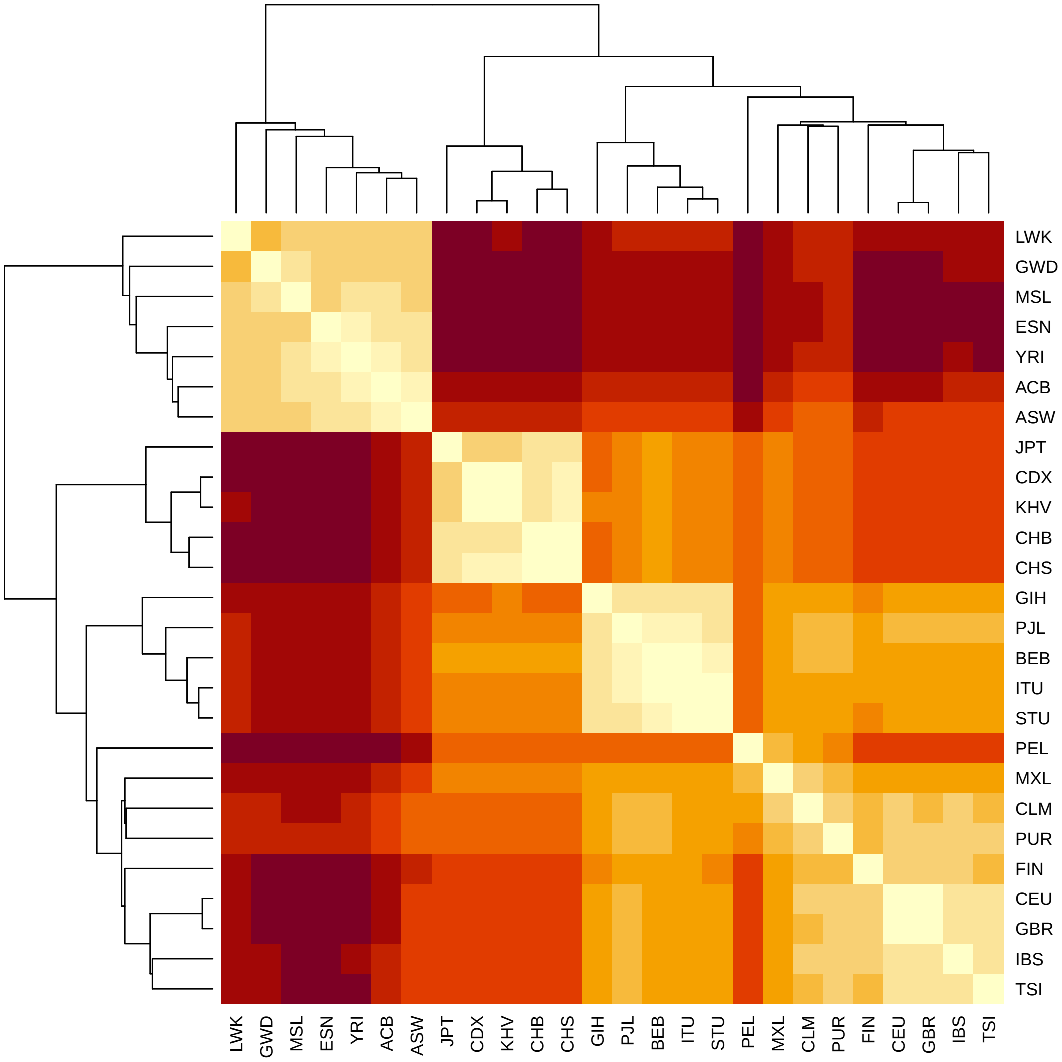


Figure SA6: Heatmap with clustering based on the Euclidean distances between centers of pairs of the 26 1000G populations.

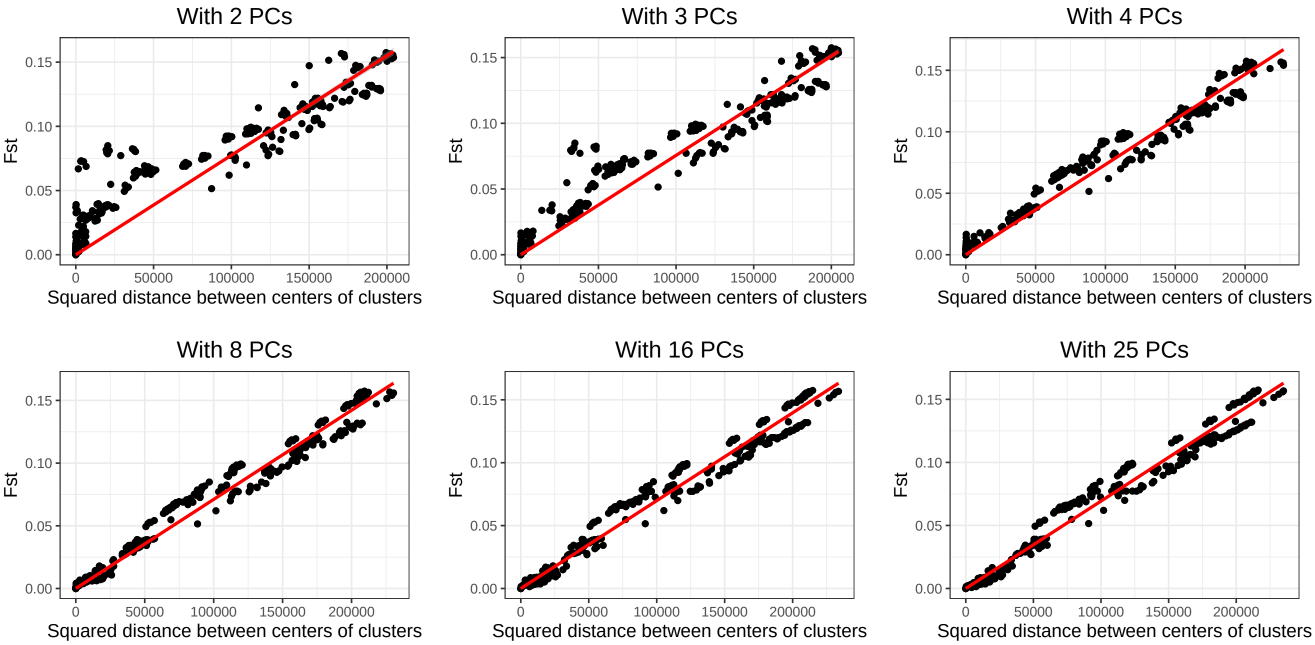


Figure SA7: Comparing F_{ST} to the squared Euclidean distances on the PCA space between centers of pairs of the 26 1000G populations. Distances are computed using different numbers of Principal Components (PCs).

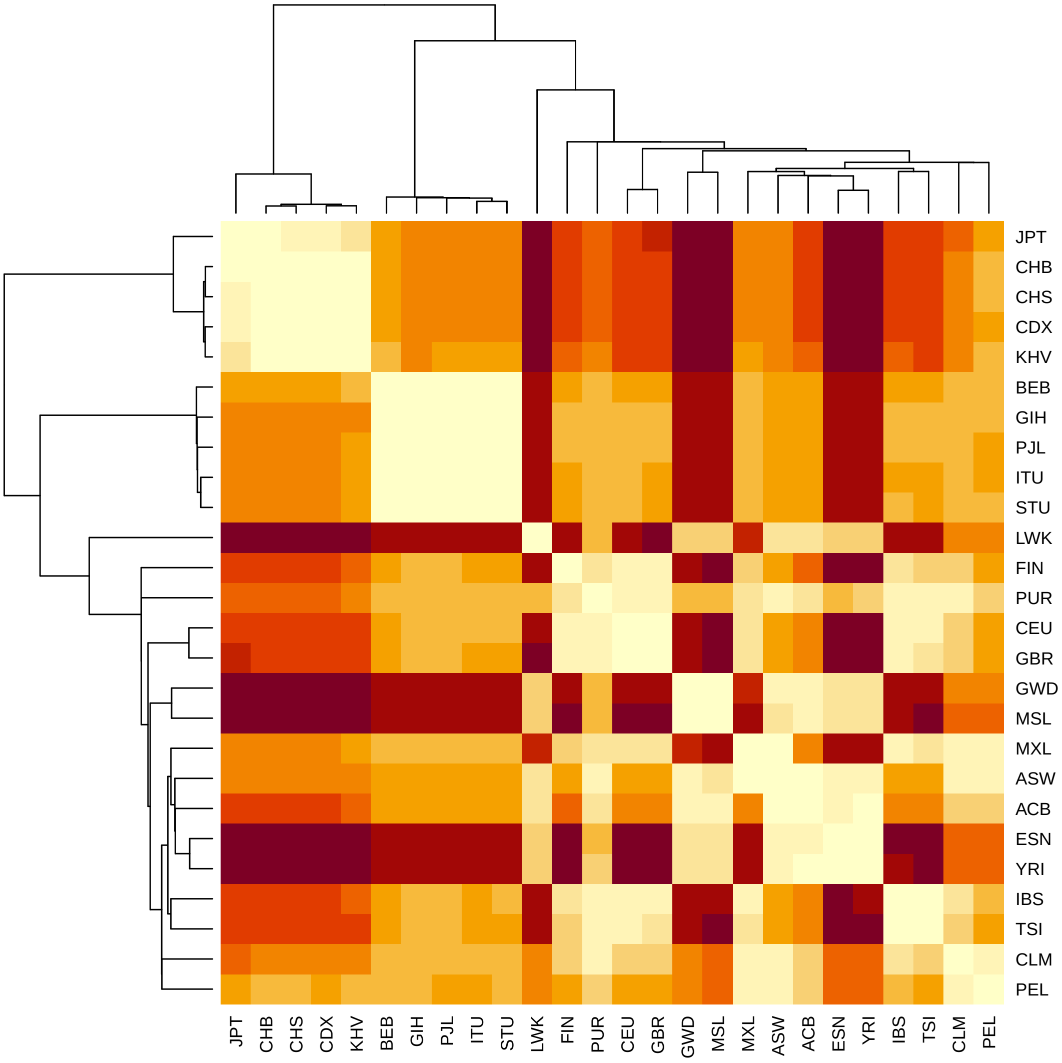


Figure SA8: Heatmap with clustering based on the shortest distances between individuals in pairs of the 26 1000G populations.

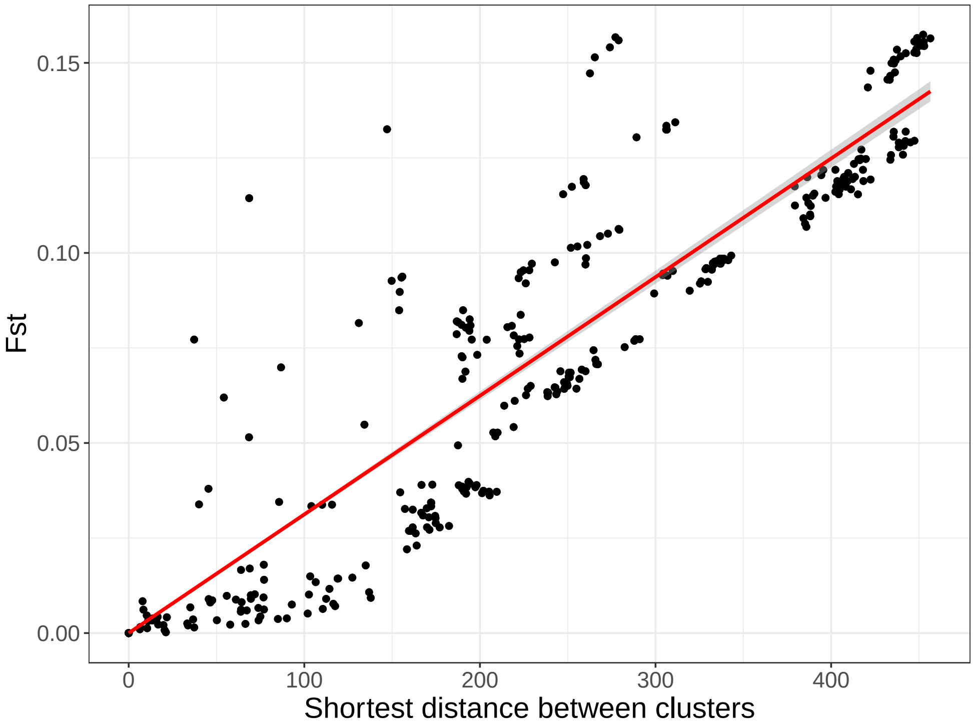


Figure SA9: Comparing F_{ST} to the shortest distances between individuals in pairs of the 26 1000G populations.

PCA-based ancestry inference

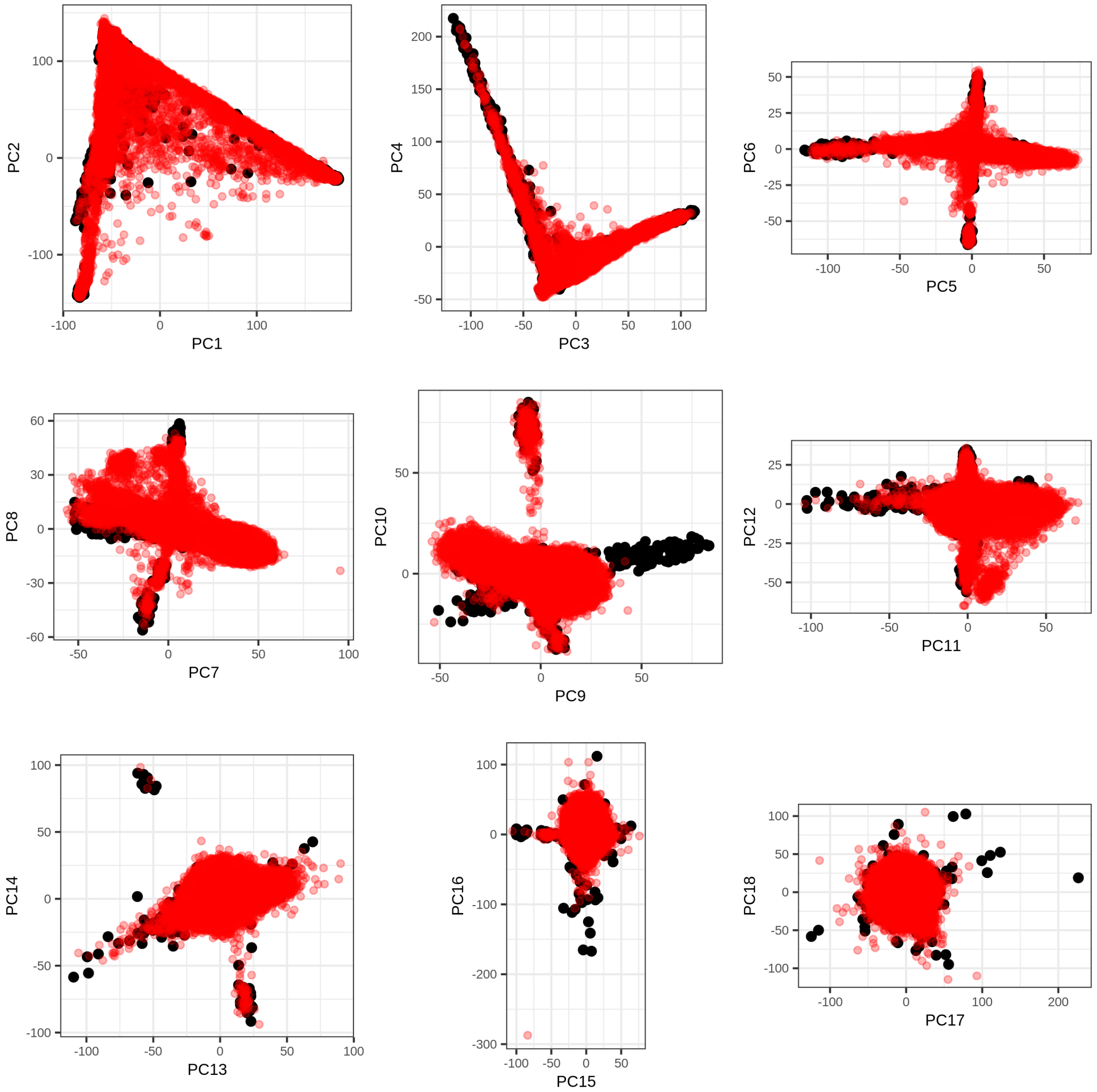


Figure SA10: First 18 PC scores of the 1000G data (in black), onto which the UK Biobank data has been projected (in red).

Table SA7: Self-reported ancestry (left) of UKBB individuals and their matching to 1000G continental populations (top) by our method. See the description of 1000G populations at `https://www.internationalgenome.org/category/population/`.

	AFR	AMR	EAS	EUR	SAS	Not matched
British	2		1	421457	6	9548
Irish				12328		425
White	1	1	1	499		43
Other White		40		11334	1	4440
Indian				5	4922	789
Pakistani					1421	327
Bangladeshi					217	4
Chinese			1453	1		50
Other Asian	1		279		939	528
Caribbean	3848				25	424
African	2633			1		570
Other Black	74				2	42
Asian or Asian British			2		20	20
Black or Black British	20			2		4
White and Black Caribbean	24	1		8	1	563
White and Black African	5			6		391
White and Asian		1	2	27	26	746
Unknown	835	173	576	2296	633	3307

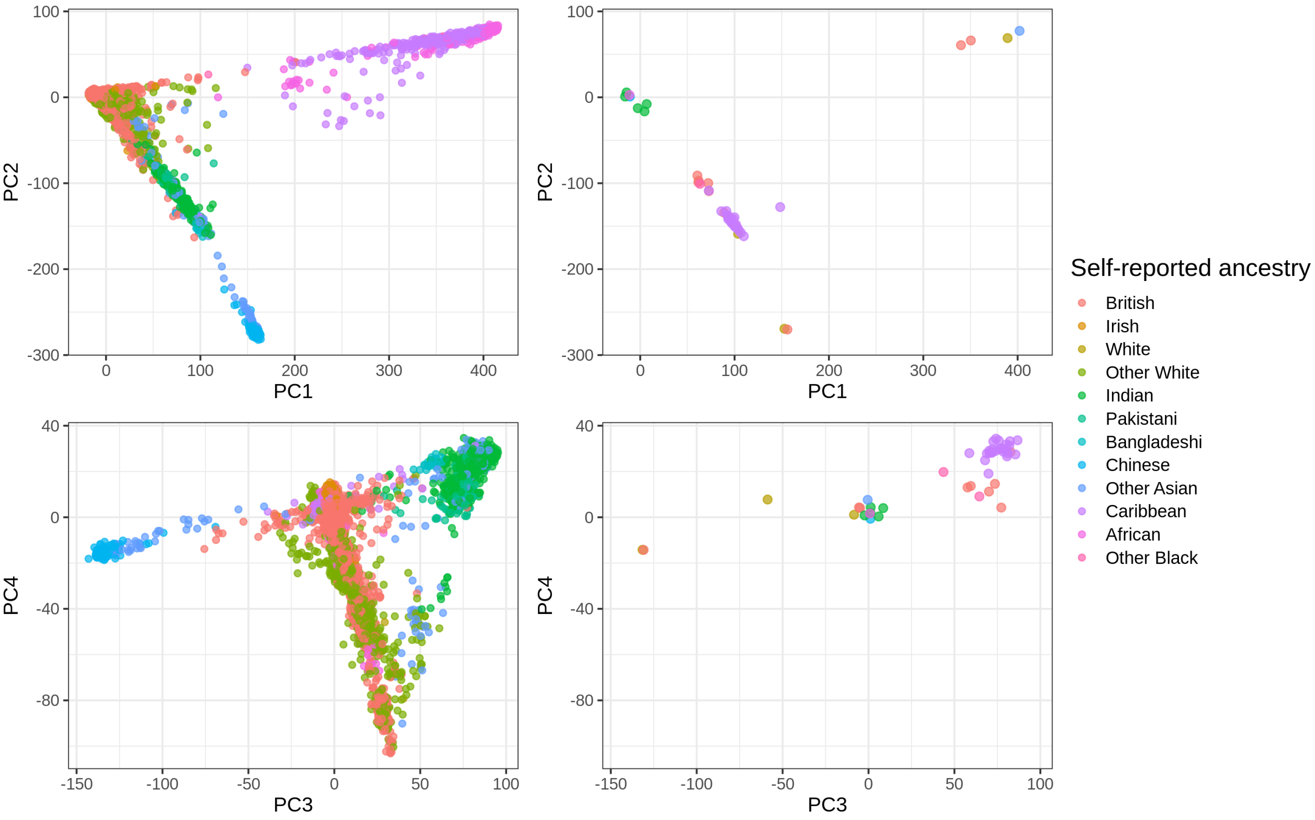


Figure SA11: PC scores (computed in the UK Biobank) colored by self-reported ancestry. On the left, these are 50,000 random individuals. On the right, these are the 47 individuals with some discrepancy between their self-reported-ancestry and our ancestry estimation (see table SA7).

Table SA8: Self-reported ancestry (left) of UKBB individuals and their matching to 1000G continental populations (top) using 20-wNN. See the description of 1000G populations at `https://www.internationalgenome.org/category/population/`.

	AFR	AMR	EAS	EUR	SAS	Not matched
British	4	50	6	430696	95	163
Irish				12748	3	2
White	1	2	1	540	1	
Other White		170	1	15533	18	93
Indian				21	5680	15
Pakistani				3	1742	3
Bangladeshi					220	1
Chinese		7	1483	3	3	8
Other Asian	1	1	359	216	1138	32
Caribbean	4117	1			36	143
African	3000		1	2	2	199
Other Black	90	1		1	5	21
Asian or Asian British			2	4	34	2
Black or Black British	23			2		1
White and Black Caribbean	93	16		74	11	403
White and Black African	102	13		52	4	231
White and Asian		42	10	242	349	159
Unknown	1024	541	712	3774	1020	749

Table SA9: Self-reported ancestry (top) of UKBB individuals and their matching to 1000G populations (left) by our method. See the description of 1000G populations at https://www.internationalgenome.org/category/population/.

	British	Irish	White	Other White	Indian	Pakistani	Bangladeshi	Chinese	Other Asian	Caribbean	African	Other Black	Unknown
AFR_ACB										2024	66	34	198
AFR_ASW	2									1072	31	11	134
AFR_ESN										1	270	1	47
AFR_GWD											42		9
AFR_LWK			1								284	1	69
AFR_MSL										3	144	3	23
AFR_YRI									1	748	1796	24	404
AMR_CLM				18									27
AMR_MXL				21									117
AMR_PEL			1	1									30
AMR_PUR													1
EAS_CDX								4	15				10
EAS_CHB								218	23				33
EAS_CHS	1		1					907	17				42
EAS_JPT								10	53				221
EAS_KHV								314	171				274
EUR_CEU	183646	854	181	5802	2			1					883
EUR_FIN	1			126									1
EUR_GBR	235579	11461	294	2446	3						1		1066
EUR_IBS	68		7	775									24
EUR_TSI	2163	13	17	2185									365
SAS_BEB				1	229	17	215		92	20		1	209
SAS_GIH					416								4
SAS_ITU	1				813	12			220	4			135
SAS_PJL	5				3332	1392	2		203	1		1	238
SAS_STU					132				424				94
Not matched	9548	425	43	4440	789	327	4	50	528	424	570	42	5031

Table SA10: Ancestry (left) of POPRES individuals and their matching to 1000G populations (top) by our method. See the description of 1000G populations at https://www.internationalgenome.org/category/population/.

	EUR_CEU	EUR_FIN	EUR_GBR	EUR_IBS	EUR_TSI	Not matched
Anglo-Irish Isles	136		127		2	1
Belgium	43					
Central Europe	47				8	
Eastern Europe	27				1	2
France	49		3	35	2	
Germany	67		3		1	
Italy	1			11	204	3
Netherlands	13		4			
Scandinavia	13	1	1			
SE Europe	12			3	70	9
SW Europe	1			261	1	1
Switzerland	179			32	11	

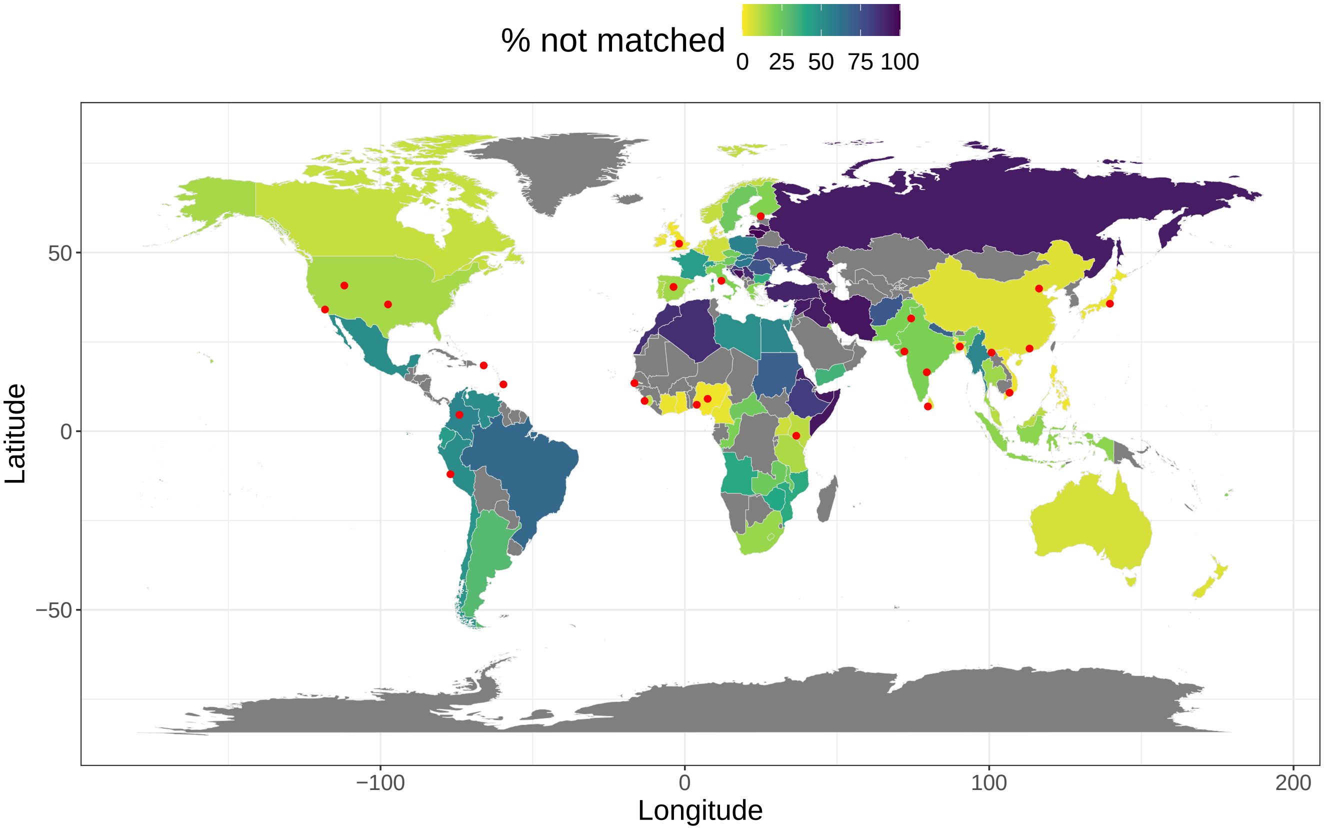


Figure SA12: Percentage of individuals from the UK Biobank that could not been matched to any of the 26 1000G populations using our method, per country of birth (Field 20115). Countries in grey contain less than 30 individuals, therefore their percentages are not represented. Red points represent the locations of the 1000G populations, accessed from `https://www.internationalgenome.org/data-portal/population`. Note that "Gujarati Indian from Houston, Texas" were manually moved to Gujarat (22.309425, 72.136230), "Sri Lankan Tamil from the UK" to Sri Lanka (6.927079, 79.861244), and "Indian Telugu from the UK" to (16.5, 79.5) to better reflect the location of their ancestors. Also note that "Utah Residents with Northern and Western European Ancestry", "Americans of African Ancestry in SW USA", "African Caribbeans in Barbados" and "Mexican Ancestry from Los Angeles USA" are probably not located at their ancestral location.

PCA-based ancestry grouping

Table SA11: Self-reported ancestry (left) of UKBB individuals and their matching to ancestry groups (top) by our method.

	British	Indian	Pakistani	Bangladeshi	Chinese	Caribbean	African	Not matched
British	426210	6	4	1	1	2		4790
Irish	12712							41
White	492				1			52
Other White	10932	1	1	1				4880
Indian	6	1764	2488	1321				137
Pakistani	1	362	1299	63				23
Bangladeshi		3		215				3
Chinese	1		1		1437			65
Other Asian	4	113	169	745	62		1	653
Caribbean		2		23		2325	1148	799
African	1		1			74	2271	857
Other Black		1	1	1		36	33	46
Asian or Asian British		7	16	3	1			15
Black or Black British	2					11	9	4
White and Black Caribbean	7			1		10	1	578
White and Black African	6					1	2	393
White and Asian	59	31	7	19				686
Unknown	2008	129	189	421	114	214	505	4240

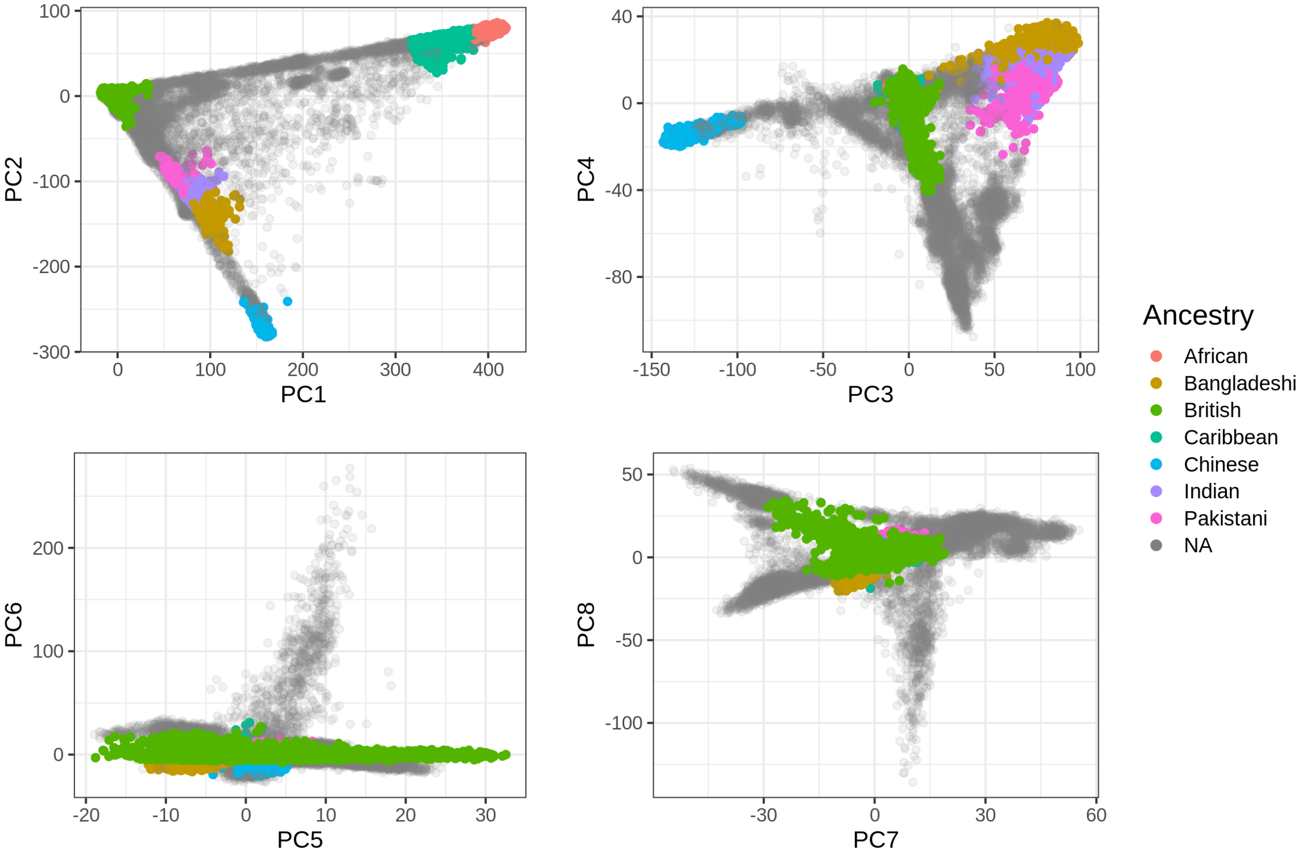


Figure SA13: The first eight PC scores computed from the UK Biobank (Field 22009) colored by the homogeneous ancestry group we infer for these individuals.

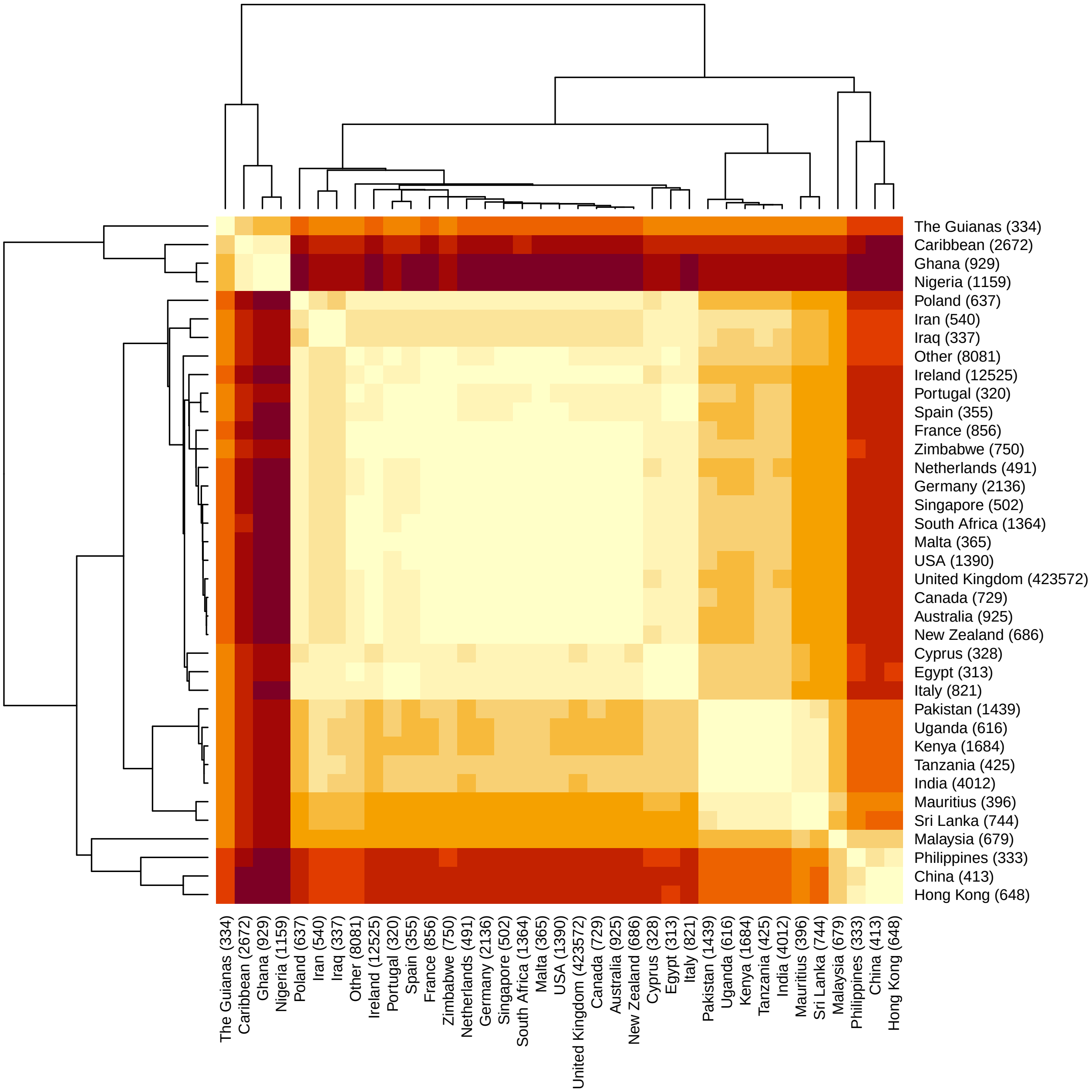


Figure SA14: Heatmap with clustering based on the distances in the PCA space between centers of pairs of the countries of birth in the UK Biobank.

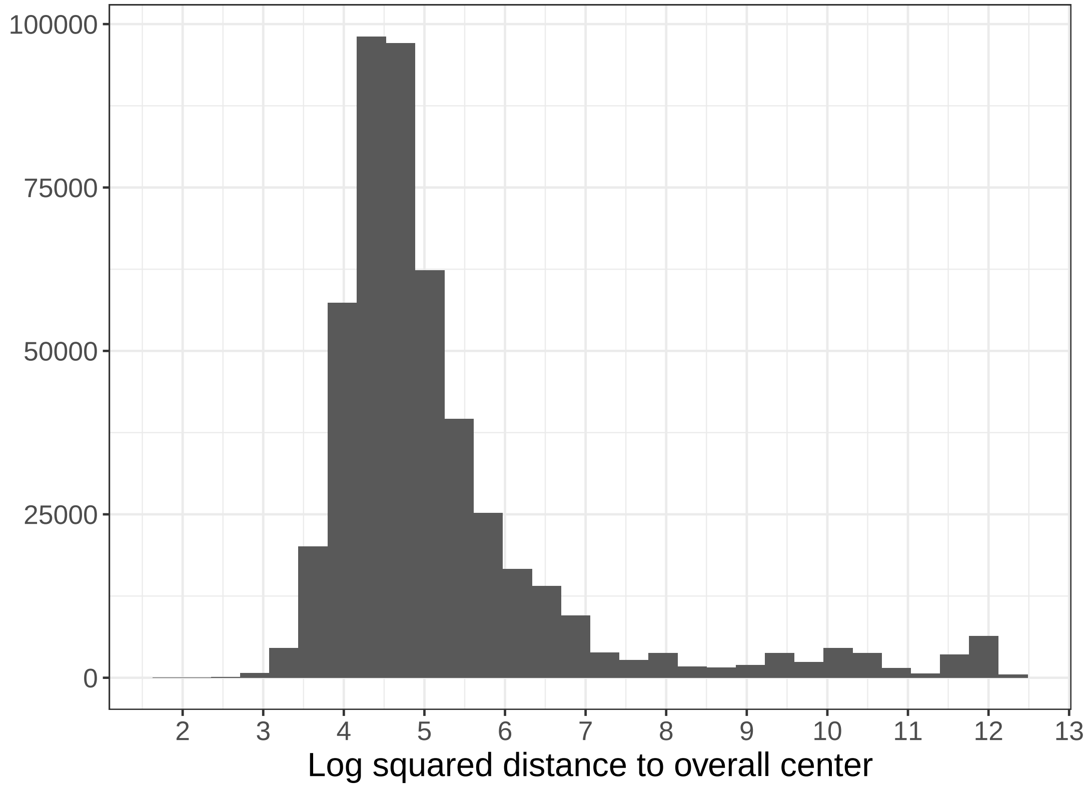


Figure SA15: Histogram of (log) squared distances from the UK Biobank PC scores to the geometric median of the all UKBB individuals. Here we use a threshold at 7, based on visual inspection. Alternatively, a more stringent threshold at 6 could also be used.

Supplementary Note: Comparison between bigstatsr and snpnet for fitting penalized regressions on very large genetic data

Penalized regression with L1 penalty, also known as "lasso", has been widely used since it proved to be an effective method for simultaneously performing variable selection and model fitting (Tibshirani 1996). R package glmnet is a popular software to fit the lasso efficiently (Friedman *et al.* 2010). However, glmnet cannot handle very large datasets such as biobank-scale data that are now available in human genetics, where both the sample size and the number of variables are very large. One strategy used to run penalized regressions on large datasets such as the UK Biobank (Bycroft *et al.* 2018) has been to apply a variable pre-selection step before fitting the lasso (Lello *et al.* 2018). Recently, authors of the glmnet package have developed a new R package, snpnet, to fit penalized regressions on the UK Biobank without having to perform any pre-filtering (Qian *et al.* 2020). Earlier, we developed two R packages for efficiently analyzing large-scale (genetic) data, namely bigstatsr and bigsnpr (Privé *et al.* 2018). We then specifically derived a highly efficient implementation of penalized linear and logistic regressions in R package bigstatsr, and showed how these functions were useful for genetic prediction with some applications to the UK Biobank (Privé *et al.* 2019). Here we benchmark bigstatsr against snpnet for fitting penalized regressions on large genetic data. Through some theoretical expectations and empirical comparisons, we show that package bigstatsr is generally much faster than snpnet. We also take that opportunity to provide more recommendations on how to fit penalized regressions in the context of genetic data.

Main motivation for snpnet

Before we can present the main motivation behind snpnet developed by Qian *et al.* (2020), let us recall how the lasso regression is fitted. Fitting the lasso consists in finding regression coefficients β that minimize the following regularized loss function

$$L(\lambda) = \underbrace{\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} X_{i,j} \beta_j \right)^2}_{\text{Loss function}} + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{\text{Penalization}} , \tag{SS1}$$

where X denotes the matrix composed of p (standardized) genotypes and possible covariates (e.g. sex, age and principal components) for n individuals, y is the (continuous) trait to predict, λ (> 0) is a regularization hyper-parameter that control the strength of the penalty. For a sequence of λ, one can find $\mathrm{argmin}_\beta \, L(\lambda)$ using cyclical coordinate descent (Friedman *et al.* 2010). To speed up the coordinate descent, one can use sequential strong rules for discarding lots of variables, i.e. setting lots of β_j to 0, a priori (Tibshirani *et al.* 2012). Therefore the cyclical coordinate descent used to solve the lasso can be performed in a subset of the data only thanks to these strong rules. However, the main drawback of these strong rules is that they require checking Karush-Kuhn-Tucker (KKT) conditions a posteriori, usually in two phases. These KKT conditions are first checked in the ever-active set, i.e. the set of all variables j with $\beta_j \neq 0$ for any previous λ. Then, the cyclical coordinate descent has to be rerun while adding the new variables that do not satisfy these KKT conditions (if any). In a second phase, the KKT conditions are also checked for all the remaining variables, i.e. the ones not in the ever-active set. This last step requires to pass over the whole dataset at least once again for every λ tested. Then, when the available random access memory (RAM) is not large enough to cache the whole dataset, data has to be read from disk, which can be extremely time consuming. To alleviate this particular issue, Qian *et al.* (2020) have developed a clever approach called batch screening iterative lasso (BASIL) to be able to check these KKT conditions on the whole dataset only after having fitted solutions for many λ, instead of performing this operation for each λ. Hence, for very large datasets, the BASIL procedure enables to fit the exact lasso solution faster than when checking the KKT conditions for all variables at each λ, as performed in e.g. R package biglasso (Zeng and Breheny 2017).

A more pragmatic approach in bigstatsr

In our R package bigstatsr, we proposed a different strategy. We also check the KKT conditions for variables in the ever-active set, i.e. for a (small) subset of all variables only; this first checking is therefore fast. However, KKT conditions almost always hold when $p > n$ (Tibshirani *et al.* 2012), which is particularly the case for the remaining variables in the second phase of checking. Because of this, we decided in Privé *et al.* (2019) to skip this second checking when designing functions `big_spLinReg` and `big_spLogReg` for fitting penalized regression on very large datasets in R package bigstatsr. Thanks to this approximation, these two functions effectively access all variables only once at the very beginning to compute the statistics used by the strong rules, and then access a subset of variables only (the ever-active set). As we show later, this means that fitting penalized regressions using the approximation we proposed in Privé *et al.* (2019) is computationally more efficient than using the BASIL procedure proposed by Qian *et al.* (2020), and yet provides equally accurate predictors. Moreover, as bigstatsr uses memory-mapping, data that resides on disk is accessed only once from disk to memory and then stays in memory while there is no need to free memory. Only when the ever-active set becomes very large, e.g. for very polygenic traits, memory can become an issue, but this extreme case would become a problem for package snpnet as well. Please refer to the Discussion section of Privé *et al.* (2019) for more details on these matters. In summary, bigstatsr effectively performs only one pass on the whole dataset while snpnet performs many passes, even though the number of passes in snpnet is reduced thanks to the

BASIL approach. Moreover, bigstatsr still uses a single pass even when performing CMSA (a variant of cross-validation (CV), see figure SS1) internally, whereas performing CV with snpnet would multiply the number of passes to the data by the number of folds used in the CV.

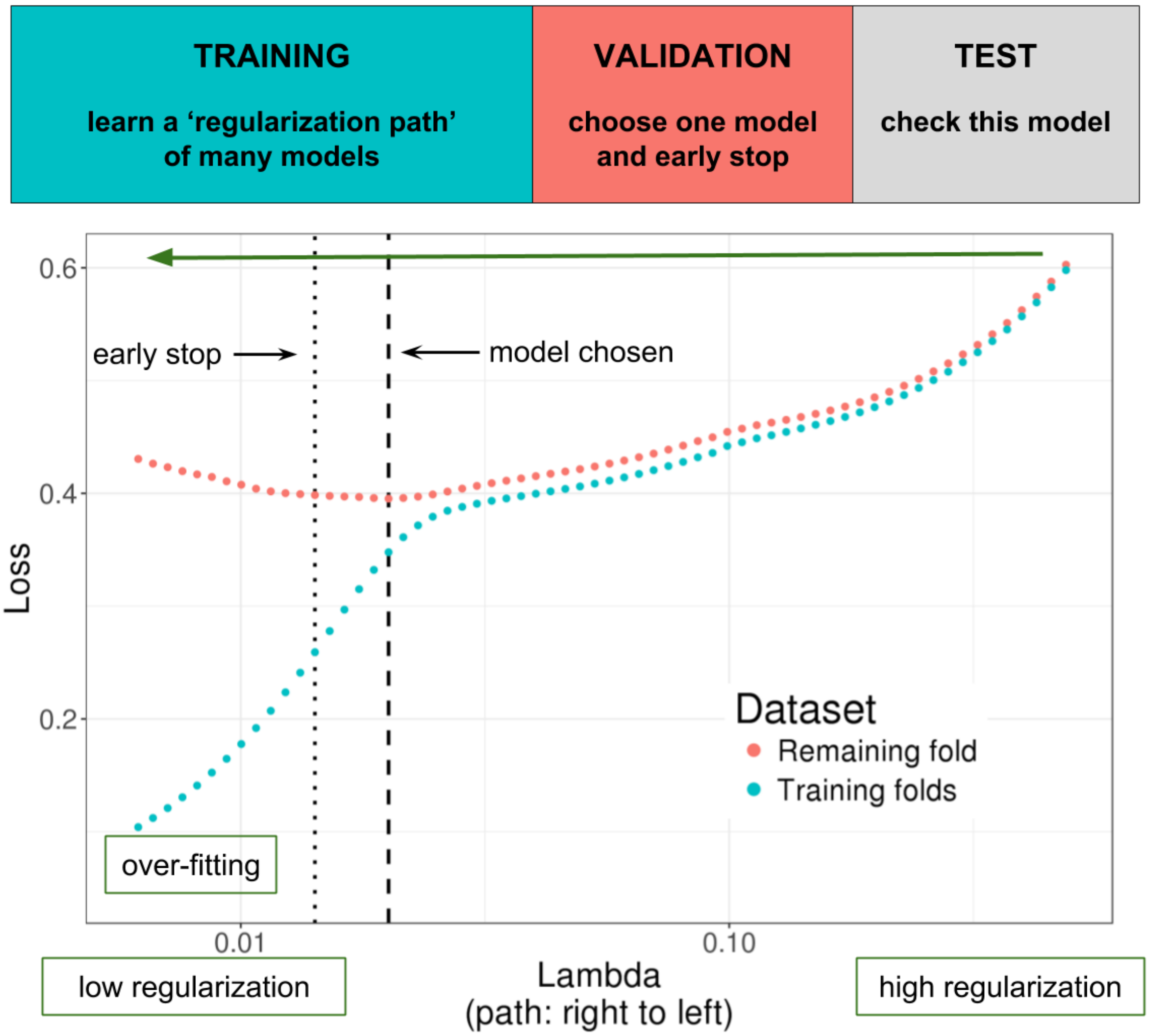


Figure SS1: Illustration of one turn of the Cross-Model Selection and Averaging (CMSA) procedure. This figure comes from Privé *et al.* (2019); the Genetics Society of America has granted us permission to re-use it. First, this procedure separates the training set in K folds (e.g. 10 folds). Secondly, in turn, each fold is considered as an inner validation set (red) and the other $(K - 1)$ folds form an inner training set (blue). A "regularization path" of models is trained on the inner training set and the corresponding predictions (scores) for the inner validation set are computed. The model that minimizes the loss on the inner validation set is selected. Finally, the K resulting models are averaged; this is different to standard cross-validation where the model is refitted on the whole training set using the best-performing hyper-parameters. We also use this procedure to derive an early stopping criterion so that the algorithm does not need to evaluate the whole regularization paths, making this procedure much faster.

Benchmark

Before, we have presented why we expect bigstatsr to be more efficient than snpnet. To practically support this claim, we perform comparisons for the four real traits used in the UK Biobank analyses of Qian *et al.* (2020). We compare R package snpnet (v0.3) with bigstatsr (v1.3) and bigsnpr (v1.5). We use similar quality controls as Qian *et al.* (2020) (see "Data & Methods"). We also use the same splitting strategy: 20% test, 20% validation and 60% training. To use the same sets for bigstatsr as for snpnet, we use the same test set, use K=4 folds for training with bigstatsr while making sure the first split is composed of the same 20% of the data used for validation in snpnet. Moreover, we use penalty factors to effectively use unscaled genotypes in bigstatsr (see "Conclusion & further recommendations"), as performed by default in snpnet. This enables us to compare predictions from snpnet and bigstatsr using the exact same model and the same single validation fold. Note that, to make the most of the training set, bigstatsr uses CMSA (Figure SS1) while Qian *et al.* (2020) propose to refit the model (on the whole training + validation) using the best λ identified using the validation set in snpnet. Also note that the parallelism used by snpnet and bigstatsr is different; snpnet relies on PLINK 2.0 to check KKT conditions in parallel, while bigstatsr parallelizes fitting of models from different folds and hyper-parameters. Because bigstatsr uses memory-mapping, the data is shared across processes and therefore it can fit these models in parallel without multiplying the memory needed. We allow for 16 cores to be used in these comparisons; bigstatsr effectively uses only 4 here (the number of folds). We allow for 128 GB of RAM for bigstatsr, but allow for 500 GB of RAM for snpnet because we had memory issues running it with only 128 GB or 256 GB.

Table SS1 presents the results of this benchmark. Fitting lasso is 35 times faster using bigstatsr than using snpnet for high cholesterol, 29 times faster for asthma, 16 times faster for BMI, and 4.5 times faster for height. When using only one validation fold for choosing the best-performing λ and no refitting, snpnet and bigstatsr provide the same predictive performance, validating the use of the approximation in bigstatsr. When using the whole training set, i.e. when refitting in snpnet and using CMSA in bigstatsr, predictive performance is much higher than when the validation set is not used for training. For example, partial correlation for height is of 0.6116 with CMSA (i.e. using the average of 4 models) compared to 0.5856 when using only one of these models, showing how important it is to make the most of the training + validation sets. Also, CMSA can provide slightly higher predictive performance than the refitting strategy in snpnet, with e.g. a partial correlation of 0.3324 vs 0.3221 for BMI.

Conclusion & further recommendations

We have found the BASIL approach derived in Qian *et al.* (2020) to be a clever approach that alleviates the I/O problem of other penalized regression implementations for very large datasets. BASIL makes significant and valuable contributions to the important problem of fitting penalized regression models efficiently. However, we also find that the implementation of BASIL in snpnet is still an order of magnitude slower than our package bigstatsr, which uses a simpler and more pragmatic approach (Privé *et al.* 2019). Hereinafter we also come back to some statements made in Qian *et al.* (2020) and provide more recommendations on how to best use

Table SS1: Benchmark of snpnet against bigstatsr in terms of predictive performance and computation time. Predictive performance is reported in terms of partial correlations between the polygenic scores and the phenotypes, residualized using the covariates. Timings are reported in minutes. Timings for snpnet report the training for 60% of the data (using the training set only) + the refitting for 80% of the data (using both the training and validation sets). Timings for bigstatsr report the time taken by the CMSA procedure (fitting K=4 models here).

	snpnet			bigstatsr		
Trait	Perf. (1 fold)	Perf. (refit)	Time	Perf. (1 fold)	Perf. (CMSA)	Time
Asthma	0.1349	0.1438	188 + 101	0.1348	0.1493	10
High cholesterol	0.1254	0.1366	101 + 146	0.1257	0.1387	7
BMI	0.3031	0.3231	161 + 893	0.3018	0.3324	65
Height	0.5871	0.6106	409 + 715	0.5856	0.6116	249

penalized regression for deriving polygenic scores based on very large individual-level genetic data. This also enables us to highlight further similarities and differences between implementations from snpnet and bigstatsr.

First, in their UK Biobank applications, Qian *et al.* (2020) have tried using elastic-net regularization (a combination of L1 and L2 penalties) instead of lasso (only L1), i.e. introducing a new hyper-parameter α ($0 < \alpha < 1$, with the special case of $\alpha = 1$ being the L1 regularization). They show that L1 regularization is very effective for very large sample sizes, and elastic-net regularization is not needed in this case, which we have also experienced. Yet, in smaller sample sizes and for very polygenic architectures, we showed through extensive simulations that using lower values for α can significantly improve predictive performance (Privé *et al.* 2019). In Qian *et al.* (2020), they tried $\alpha \in \{0.1, 0.5, 0.9, 1\}$; we recommend to use a grid on the log scale with smaller values (e.g. 1, 0.1, 0.01, 0.001, and even until 0.0001) for smaller sample sizes. Note that using a smaller α leads to a larger number of non-zero variables and therefore more time and memory required to fit the penalized regression. In functions `big_spLinReg` and `big_spLogReg` of R package bigstatsr, we allow to directly test many α values in parallel within the CMSA procedure. Therefore an optimal α value can be chosen automatically within the CMSA framework, without the need for more passes on the data.

Second, for large datasets, one should always use early-stopping. We have not found this to be emphasized enough in Qian *et al.* (2020). Indeed, while fitting the regularization path of decreasing λ values on the training set, one can monitor the predictive performance on the validation set, and stop early in the regularization path when the model starts to overfit (Figure SS1). For large datasets, performance on the validation sets is usually very smooth and monotonic (before and after the minimum) along the regularization path, then one can safely stop very early, e.g. after the second iteration for which prediction becomes worse on the validation set. This corresponds to setting `n.abort=2` in bigstatsr and `stopping.lag=2` in snpnet. This is particularly useful because, when we move down the regularization path of λ values, more and more variables enter the model and the cyclical coordinate descent takes more and more time and memory. Therefore, the early-stopping criterion used in both bigstatsr and snpnet prevents from fitting very costly models, saving a lot of time and memory.

Third, Qian *et al.* (2020) recommend not to use scaled genotypes when applying lasso to genetic data.

However, using scaled genotypes is common practice in genetics, and is the assumption behind models in popular software such as GCTA and LDSC (Yang *et al.* 2011; Bulik-Sullivan *et al.* 2015). Scaling genotypes assumes that, on average, all variants explain the same amount of variance and that low-frequency variants have larger effects. Speed *et al.* (2012) argued that this assumption might not be reasonable and proposed another model: $\mathbb{E}[h_j^2] \propto [p_j(1-p_j)]^\nu$, where h_j^2 is the variance explained by variant j and p_j is its allele frequency. In Speed *et al.* (2017), they estimated ν to be between -0.25 and -0.5 for most traits. Note that scaling genotypes by dividing them by their standard deviations SD_j as done by default in bigstatsr assumes $\nu = -1$ while not using any scaling as argued by Qian *et al.* (2020) assumes $\nu = 0$. Therefore, using a trade-off between these two approaches can provide higher predictive performance and is therefore recommended (Zhang *et al.* 2020). In the case of L1 regularization, using a different scaling can be obtained by using different penalty factors λ_j in equation (SS1), which is an option available in both bigstatsr and snpnet. For example, using $\lambda_j = 1/\text{SD}_j$ allows to effectively use unscaled genotypes. Recently, we have implemented a new parameter `power_scale` to allow for different scalings when fitting the lasso in bigstatsr. Note that a vector of values to try can be provided, and the best-performing scaling is automatically chosen within the CMSA procedure.

Fourth, Qian *et al.* (2020) stated that bigstatsr "do not provide as much functionality as needed in [their] real-data application", mainly because bigstatsr requires converting the input data and cannot handle missing values. It is true that bigstatsr uses an intermediate format, which is a simple on-disk matrix format accessed via memory-mapping. However, package bigsnpr provides fast parallel functions `snp_readBed2` for converting from '.bed' files and `snp_readBGEN` for converting from imputed '.bgen' files, the two formats used by the UK Biobank. For example, it took 6 minutes only to read from the UK biobank '.bed' file used in this paper. We then used function `snp_fastImputeSimple` to impute by the variant means in 5 minutes only, which is also the imputation strategy used in snpnet. When reading imputed dosages instead, it takes less than one hour to access and convert 400K individuals over 1M variants using function `snp_readBGEN` with 15 cores, and less than three hours for 5M variants. When available, we recommend to directly read from '.bgen' files to get dosages from external reference imputation. As for package snpnet, it uses the PLINK 2.0 '.pgen' format, which is still under active development (in alpha testing, see `https://www.cog-genomics.org/plink/2.0/formats#pgen`). This format is not currently provided by the UK Biobank, and can therefore be considered as an intermediate format as well.

Data & Methods

As in Qian *et al.* (2020), we use the UK Biobank data (Bycroft *et al.* 2018), which is a large cohort of half a million individuals from the UK, for which we have access to both genotypes and multiple phenotypes (`https://www.ukbiobank.ac.uk/`). We apply some quality control filters to the genotyped data; we remove individuals with more than 10% missing values, variants with more than 1% missing values, variants having a minor allele frequency < 0.01, variants with P-value of the Hardy-Weinberg exact test < 10^{-50}, and non-autosomal variants. We restrict individuals to the ones used for computing the principal components in the UK Biobank (Field 22020); these individuals are unrelated and have passed some quality control (Bycroft *et al.* 2018). We also restrict to the "White British" group defined by the UK Biobank (Field 22006) to get a set of genetically homogeneous individuals. These filters result in 337,475 individuals and 504,139 genotyped variants.

We use the same four phenotypes as used in Qian *et al.* (2020), namely height, body mass index (BMI), high cholesterol and asthma. We define height using field 50, BMI using field 21001, high cholesterol using field 20002 ("Non-cancer illness code, self-reported"). Asthma is defined using field 20002 as well as fields 40001, 40002, 41202 and 41204 (ICD10 codes); please see code for further details at `https://github.com/privefl/paper2-PRS/tree/master/response-snpnet/code`. For height and BMI, L1-penalized linear regressions are fitted using function `big_spLinReg` from bigstatsr and using parameter `family="gaussian"` in snpnet. For high cholesterol and asthma, L1-penalized logistic regressions are fitted using function `big_spLogReg` from bigstatsr and using parameter `family="binomial"` in snpnet. We use sex (Field 22001), age (Field 21022), and the first 16 principal components (Field 22009) as unpenalized covariates when fitting the lasso models.

References

Bulik-Sullivan, B. K., Loh, P.-R., Finucane, H. K., Ripke, S., Yang, J., Patterson, N., Daly, M. J., Price, A. L., Neale, B. M., of the Psychiatric Genomics Consortium, S. W. G., *et al.* (2015). LD score regression distinguishes confounding from polygenicity in genome-wide association studies. *Nature Genetics*, **47**(3), 291.

Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L. T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., *et al.* (2018). The UK biobank resource with deep phenotyping and genomic data. *Nature*, **562**(7726), 203–209.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, **33**(1), 1.

Lello, L., Avery, S. G., Tellier, L., Vazquez, A. I., de los Campos, G., and Hsu, S. D. (2018). Accurate genomic prediction of human height. *Genetics*, **210**(2), 477–497.

Privé, F., Aschard, H., Ziyatdinov, A., and Blum, M. G. (2018). Efficient analysis of large-scale genome-wide data with two R packages: bigstatsr and bigsnpr. *Bioinformatics*, **34**(16), 2781–2787.

Privé, F., Aschard, H., and Blum, M. G. (2019). Efficient implementation of penalized regression for genetic risk prediction. *Genetics*, **212**(1), 65–74.

Qian, J., Tanigawa, Y., Du, W., Aguirre, M., Chang, C., Tibshirani, R., Rivas, M. A., and Hastie, T. (2020). A fast and scalable framework for large-scale and ultrahigh-dimensional sparse regression with application to the UK Biobank. *PLOS Genetics*, **16**(10), 1–31.

Speed, D., Hemani, G., Johnson, M. R., and Balding, D. J. (2012). Improved heritability estimation from genome-wide SNPs. *The American Journal of Human Genetics*, **91**(6), 1011–1021.

Speed, D., Cai, N., Johnson, M. R., Nejentsev, S., and Balding, D. J. (2017). Reevaluation of SNP heritability in complex human traits. *Nature Genetics*, **49**(7), 986–992.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**(1), 267–288.

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., and Tibshirani, R. J. (2012). Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **74**(2), 245–266.

Yang, J., Lee, S. H., Goddard, M. E., and Visscher, P. M. (2011). GCTA: a tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*, **88**(1), 76–82.

Zeng, Y. and Breheny, P. (2017). The biglasso package: A memory- and computation-efficient solver for lasso model fitting with big data in R. *arXiv preprint arXiv:1701.05936*.

Zhang, Q., Privé, F., Vilhjalmsson, B. J., and Speed, D. (2020). Improved genetic prediction of complex traits from individual-level data or summary statistics. *bioRxiv*.