Deep Learning Models for Polygenic Risk Score on Simulated Data

Ksenia Koshkina

October 2023

1 Introduction

Despite significant advances in diagnosis and treatment, medicine does not always take a personalized approach to patients. Instead of considering each person's individual characteristics, standard methods of diagnosis and treatment are used, which are not always effective. For example, when treating multifactorial diseases such as diabetes or hypertension, doctors often use identical treatment methods for all patients, without taking into account the individual characteristics of each organism. This can lead to misdiagnosis, ineffective treatment, and undesirable side effects.

One of the main approaches in personalized medicine is polygenic risk scoring (PRS). This method allows assessing the likelihood of a patient developing a specific disease based on their genetic data. To calculate polygenic risk scores, information is used about the presence of certain genetic variants (single nucleotide polymorphisms, or SNPs) in the patient that are associated with an increased risk of developing the disease. Polymorphisms are identified using GWAS studies, which allow the relationship between genetic variants and the presence of certain diseases or phenotypic traits to be determined. During a GWAS study, the genomes of a large number of people are analyzed to identify the most significant genetic variants that may be associated with the risk of developing certain diseases or phenotypic traits. PRS assessment can be used to develop an individualized treatment and disease prevention plan for a patient.

There are many different methods for calculating PRS, and one of these methods involves machine learning and deep learning algorithms. To date, numerous studies have been conducted that demonstrate impressive results using this approach to predict polygenic risk.

An indisputable factor in predicting polygenic risk is the population for which the polygenic risk is calculated. Depending on the population, polygenic risk can take on variable values, since the genotype of each population differs. For example, causal polymorphisms for the European population may differ from causal polymorphisms for the African population, and in such a case, the application of polygenic risk is not representative and may be erroneous.

This work is devoted to predicting polygenic risk assessment on simulated data using deep learning. It is worth noting that we have expanded our research somewhat to include population analysis. After reviewing the available literature, we found no studies with a similar approach. Deep learning is more commonly used in predicting polygenic risk on datasets that include a single population. Based on this factor, we decided to combine these approaches and analyze the results.

To summarize, we can highlight the following research objectives:

1. Conduct a literature review to confirm the problem of population stratification in PRS prediction.
2. Study modern methods for solving this problem.
3. Conduct a literature review on deep learning methods and determine which architectures prevail, as well as the advantages and disadvantages of each architecture.
4. Compile a representative dataset that includes genotype-phenotype associations and consists of three populations: European, African, and Asian.
5. Train machine learning and deep learning algorithms on the simulated dataset.
6. Analyze which architecture better predicted the phenotype.
7. Analyze the role of population in phenotype prediction.

2 Literature review

As mentioned earlier, deep learning was used to calculate polygenic risk scores, but without taking population into account. However, the scientific community has conducted research to determine the role of population in predicting polygenic risk, but without the use of deep learning. In this section, we will review the main studies on which we will base our own research.

A common problem that arises when calculating polygenic risk is that most GWAS studies have been conducted on European populations, as the availability of samples from ethnic minorities is limited. This factor limits the use of such GWAS studies on other populations. [6] It is noted that an important step in preventive medicine is to expand genomic research in non-European populations. Currently, individuals of non-European descent cannot take advantage of personalized medicine due to insufficient representation in genetic studies. [3]

Epistatic interactions, phenotypic heterogeneity, or low frequency of minor alleles/non-polymorphic markers at the locus may account for differences in the prediction of different causal SNPs in different population cohorts. [1]

Solutions to this problem are proposed by the developers of the PRS-CSx tool, who are investigating methods for improving PRS prediction in genetically diverse populations. PRS-CSx is a Bayesian polygenic prediction method that combines summary GWAS statistics from multiple populations to improve the accuracy of PRS prediction in initially diverse samples. In this study, PRS-CSx is compared with existing PRS calculation methods, LDpred2 and PT (P-value thresholding).

The authors evaluated the predictive power of various polygenic prediction methods using simulation. Using the HAPGEN2 program, they simulated individual genotypes of the EUR (European), EAS (East Asian), and AFR (African) populations for HapMap3 variants. Then, they randomly selected 1

After modeling, the authors applied single-discovery methods to GWAS summary statistics obtained from 100,000 simulated samples from a European population and 20,000 samples from other countries (EAS or AFR), and evaluated their predictive performance between simulated and predicted phenotypes. The results were as follows: when the target population was European, PRS trained on European GWAS statistics were significantly more accurate than PRS trained on GWAS of non-European origin. However, when the target population was EAS or AFR, PRS trained on matched non-European GWAS were more accurate than PRS for the European dataset, even though the sample sizes of the non-European GWAS were significantly smaller. Among the three methods considered, Bayesian methods (LDpred2 and PRS-CS) outperformed PT. [10]

The second tool, SDPRX[15], a statistical method for cross-population prediction of complex traits, works in a similar way. This tool combines GWAS summary statistics and LD (Linkage Disequilibrium) matrices from two populations with effect sizes within a Bayesian model. SDPRX characterizes the joint distribution of SNP effect sizes in two populations as zero, population-specific, or with a common correlation. SDPRX takes GWAS summary statistics from two populations as input data and thus uses shared information from two populations to estimate SNP effect sizes more accurately than single-population methods such as LDpred2. [15]

With regard to deep learning algorithms in PRS calculation, the scientific community often uses this approach to predict the PRS of various diseases. Deep learning models have been used to predict the risk of developing diabetes, Alzheimer's disease, breast cancer, and other diseases. These studies have shown that neural network methods can significantly improve the accuracy of disease risk prediction compared to traditional statistical methods.

For example, deep learning models such as CNN and RNN were used to predict polygenic risk of breast cancer. The study was conducted on the NIH dbGaP database and included 26,053 cases and 23,058 controls. The authors claim that deep learning models, including CNN and RNN, outperform statistical PRS algorithms such as BLUP, BayesA, and LDpred in predicting polygenic breast cancer risk. The authors obtained ROC-AUC metrics of 67.4

Deep learning algorithms were also used to predict the obesity phenotype. In this study, GWAS results were combined with a deep learning framework to test the predictive power of statistically significant SNPs associated with the obesity phenotype. The approach demonstrates the potential of deep learning as a powerful framework for GWAS analysis, allowing information to be obtained about SNPs and important interactions between them. Statistical testing of the association between individual SNPs and obesity was performed within an additive model using logistic regression. The deep learning classifier is initialized using SNP sets with different p-values. Using a deep learning model and polymorphisms with a P-value < 0.001 (2465 SNPs), the authors obtained the following results: LogLoss=0.1150, AUC=0.9908, and MSE=0.03. [9]

A study on the use of deep learning models to predict Alzheimer's disease (AD) deserves attention. The authors evaluate the effectiveness of PRS, Lasso, and NN models for predicting the risk of developing AD based on genetic information and claim that the deep learning model classifies the risk of developing the disease more accurately than the weighted PRS and Lasso models. When classifying patients with a clinical diagnosis of AD, the best AUC-ROC score achieved by the neural network model was 0.84. The authors also find that polygenic risk for AD may correlate with pathophysiological changes in individual patients. In addition, deep learning methods allow people at risk of developing the disease to be stratified into subgroups according to their polygenic risk. This study highlights the potential of using deep learning methods to study disease mechanisms and stratify people at increased risk of developing diseases into subgroups. [14]

3 Simulation methods

Based on the studies described above, we developed the following plan for our research:

1. Simulation of the genotype of different populations using HapGen2
2. Simulation of the phenotype based on a simulated genotype dataset using PhenotypeSimulator
3. Obtaining GWAS statistics for SNPs in each population with a p-value < 0.05

We only considered simulated data for the following reasons:

1. Phenotype

To test the influence of population on polygenic risk prediction, data on a specific phenotype for several populations is required, but we do not have such data.

2. Data set quality

The performance of deep learning models depends on the quality of the data set. In a real data set, there are mostly missing values, which can reduce the number of common SNPs between both populations, while in simulated data, the number of missing SNP values is lower.

3.1 Simulation of the genotype of different populations using HapGen2

HAPGEN2[12] is a program that simulates case-control datasets based on SNP markers, based on the Lee and Stevens model [7]. The program allows simulating multiple SNPs on a single chromosome, assuming that each SNP acts independently and is in Hardy-Weinberg equilibrium.

The approach can work with markers in non-equilibrium linkage disequilibrium (LD) and simulate data sets over large regions, such as entire chromosomes. It models haplotypes based on a reference set of population haplotypes and an estimate of the recombination rate in the region, so that the simulated data has the same LD characteristics as the real data. The availability of HapMap3[3] data allows HAPGEN2 to simulate datasets from multiple populations.

HAPGEN2 accepts the following files as input:

1. legend file - a file in IMPUTE format. This file contains the following columns about variants: variant ID, base pair position, reference allele, alternative allele, biallelic variant type (SNP/INDEL/SV), ALT allele frequency in continental groups (AFR, EAS, EUR), ALT allele frequency in all samples.

2. hap file - phased haplotype file in IMPUTE format

3. map file - genetic map file in IMPUTE format (positions in NCBI b37 coordinates)

Using HAPGEN2[12] and HapMap3[3] data, we modeled the genotype on chromosomes 1-22, consisting of 400,000 SNPs for 5,000 individuals from each population—European, African, and Asian. The final genotype dataset consisted of 15,000 individuals.

The populations were stratified based on coordinates from the HapMap3 .sample file—a text file with sample ID, population, and continental group for individuals in the haplotype files. After obtaining the coordinates, we filtered the original .hap file by the coordinates of the three populations and fed the population-filtered .hap file into HAPGEN2. The output was genotype files in .gen and .sample formats.

After modeling the genotype, we performed a PCA analysis to demonstrate the differences between the genotypes of the modeled populations.

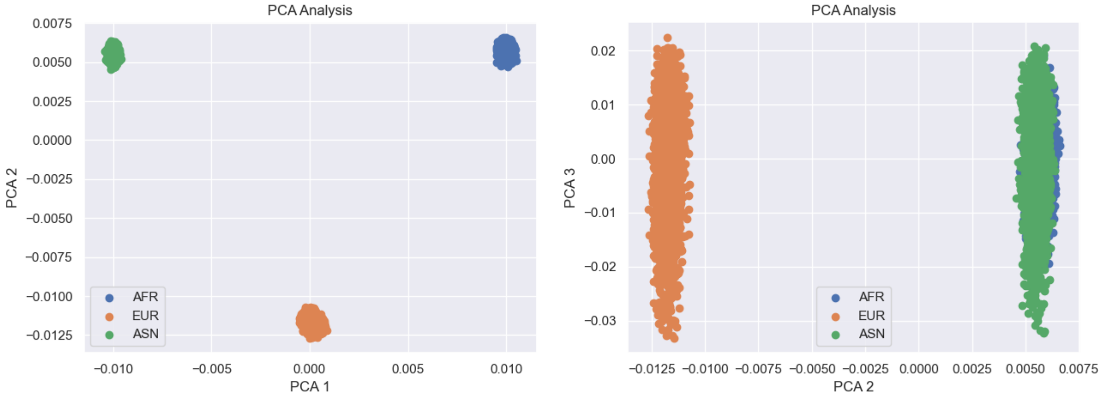


Figure 1: 2D scatter plot PCA of simulated populations. AFR - African population, EUR - European population, ASN - Asian population.

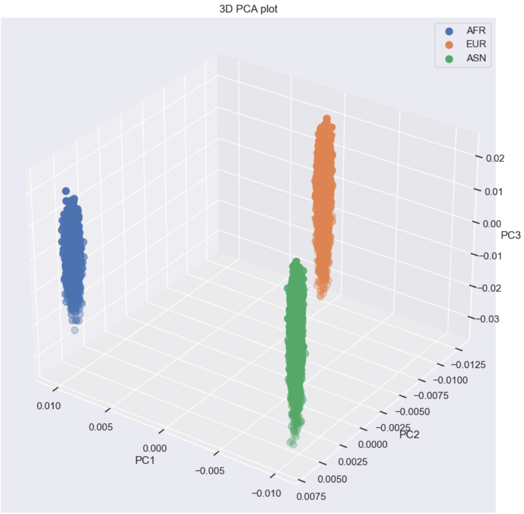


Figure 2: 3D scatter plot PCA of simulated populations. AFR - African population, EUR - European population, ASN - Asian population.

3.2 Simulation of a phenotype based on a simulated genotype using Phenotype-Simulator

PhenotypeSimulator [8] is a comprehensive phenotype simulation scheme that allows modeling of multiple phenotypic traits with several genetic loci, complex covariance structures, and observation noise.

The input to PhenotypeSimulator is a matrix simulated using HAPGEN2 (.gen and .sample files) to simulate the main genetic and non-genetic covariates, observation noise, and non-genetic correlation structures. The effect structure of the four components listed above is divided into a common effect across all traits, which allows for the creation of complex phenotype structures.

The final simulated phenotype Y is expressed as the sum of the effects of genetic variants, non-genetic covariates, correlated non-genetic effects, and observation noise effects:

$$\mathbf{Y} = \mathbf{X}^{\text{shared}}\mathbf{B}^{\text{shared}} + \mathbf{X}^{\text{ind}}\mathbf{B}^{\text{ind}} + \mathbf{W}^{\text{shared}}\mathbf{A}^{\text{shared}} + \mathbf{W}^{\text{ind}}\mathbf{A}^{\text{ind}} + \mathbf{U}^{\text{shared}} + \mathbf{U}^{\text{ind}} + \mathbf{T} + \boldsymbol{\Psi}^{\text{shared}} + \boldsymbol{\Psi}^{\text{ind}}$$

Depending on the objective, the user can select different values for the components included in the phenotype simulation. To simulate genetic effects, S random SNPs are taken from the simulated genotypes for N samples. In our study, we ran PhenotypeSimulator with default parameters, selecting a number of SNPs equal to 30,000. However, to reflect the contribution of each population's genotype, the seed parameter was fixed for each simulation session. Thus, for each population, the random SNPs were identical and had the same effect on all components.

After simulating each population, we obtained PLINK format .bed, .bim, and .fam files, which included 30,000 randomly selected identical SNPs, and a .txt file with the phenotype value.

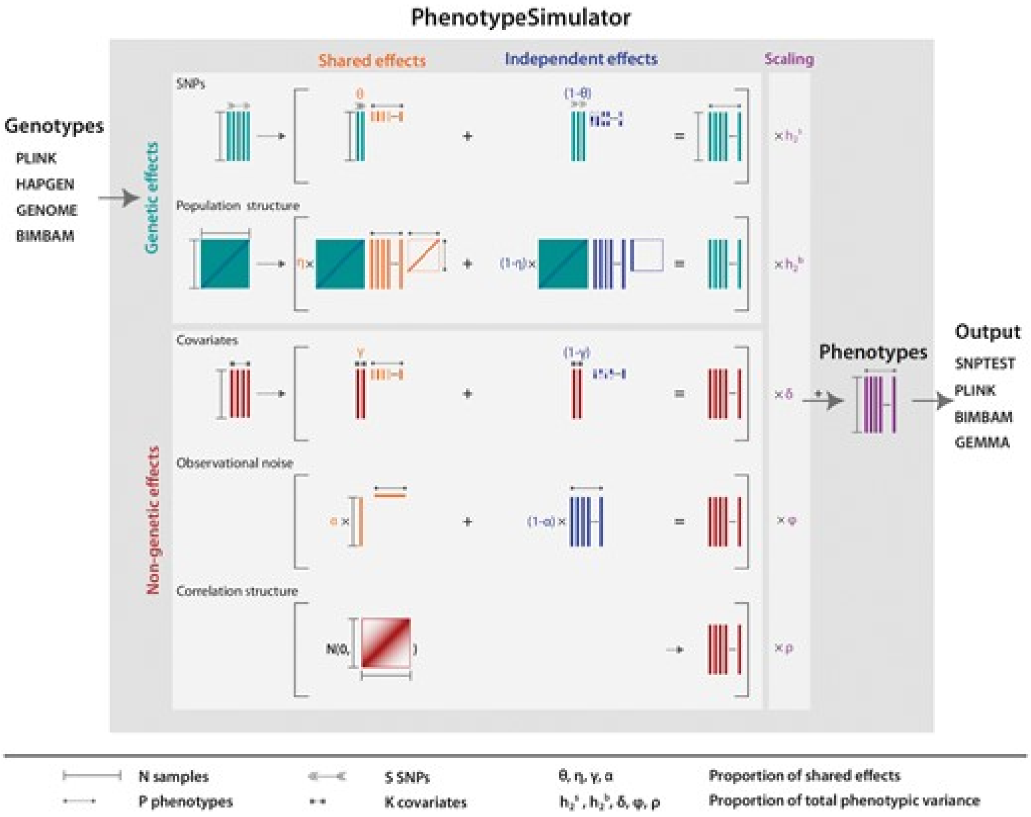


Figure 3: Figure 3. PhenotypeSimulator workflow diagram[7].

4 Data processing

4.1 P-value statistics

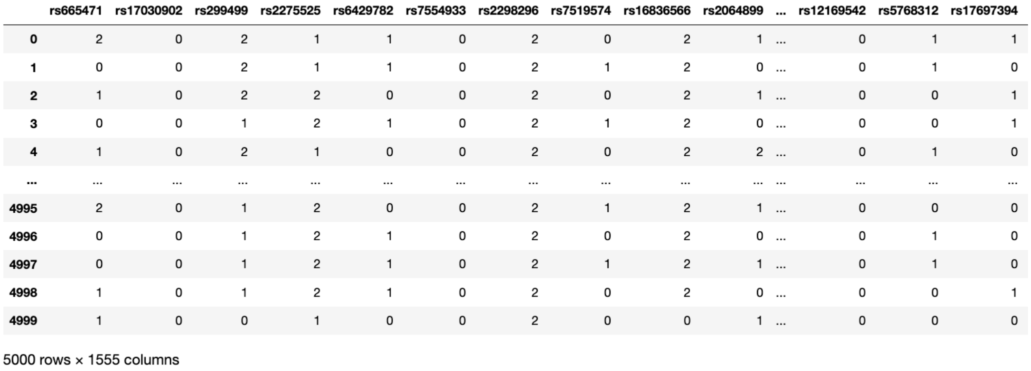
Using 30,000 SNPs for training can lead to model overfitting, so the process of pre-selecting SNPs (p-value threshold) allowed us to reduce the amount of input data and obtain only the truly important SNPs. Thus, after receiving files from PhenotypeSimulator for each population, we conducted a statistical study using a standard association test implemented in PLINK[10]. The main association test is performed for a disease trait and is based on comparing allele frequencies between cases and controls. After receiving the associated file, we filtered the SNPs by p-value, excluding SNPs with a p-value less than 0.05. The number of SNPs was different for each population. It is worth noting that statistically significant SNPs differed between populations.

4.2 Filtering the genotype file by SNP with a p-value < 0.05

Our next step was to convert the .bed, .bim, and .fam files for each population into .vcf files using PLINK. The .vcf files were then filtered for SNPs with a p-value less than 0.05.

4.3 Final data

To transfer the .vcf file to the model, we changed the homozygous and heterozygous values in the .vcf file, where '0/0' is 0, '0/1' is 1, and '1/1' is 2. Ultimately, we obtained a matrix of genotype values K x L, where K is the number of individuals—in our case, 5,000 for each population—and L is the number of statistically significant SNPs.

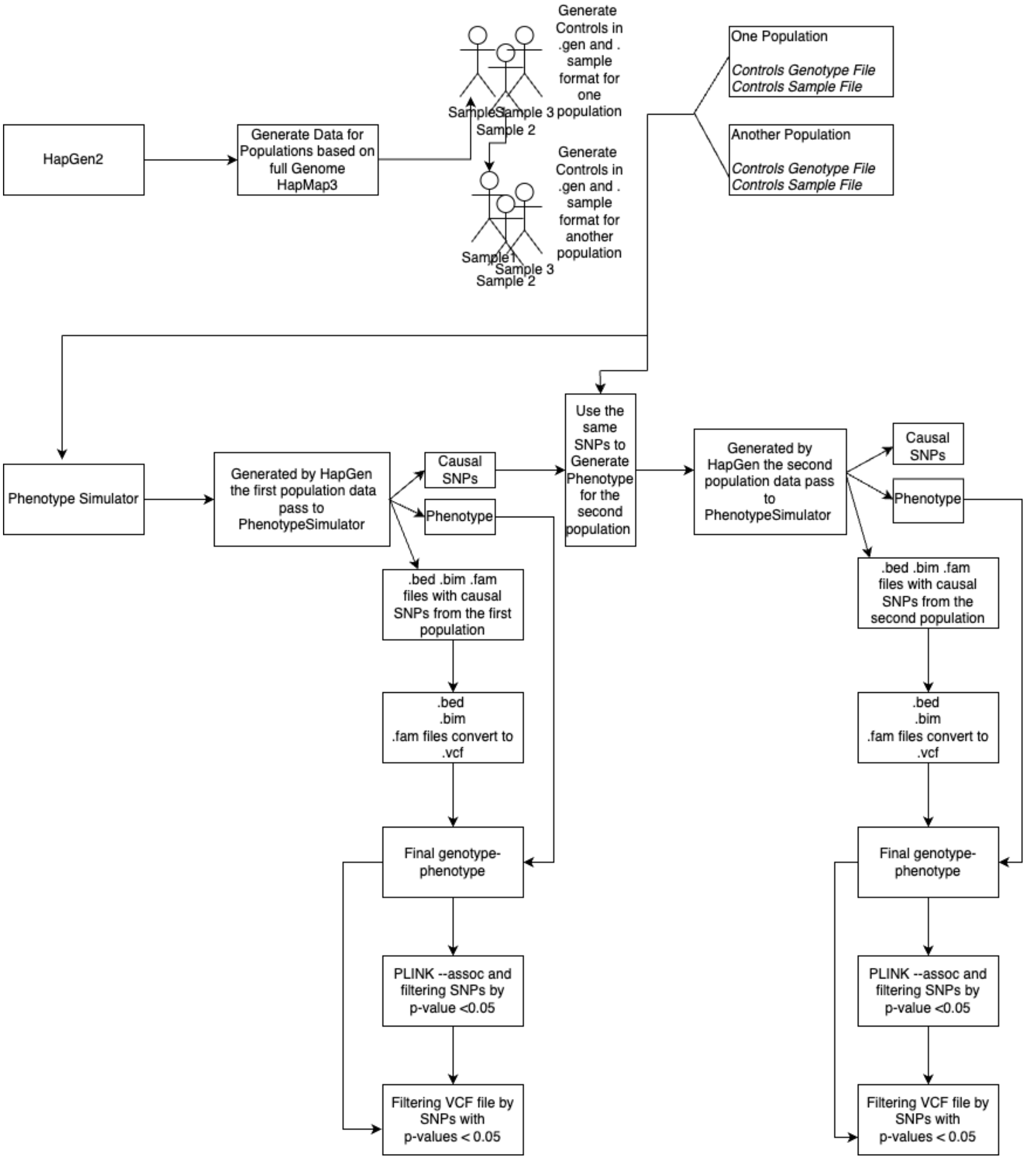


	rs665471	rs17030902	rs299499	rs2275525	rs6429782	rs7554933	rs2298296	rs7519574	rs16836566	rs2064899	...	rs12169542	rs5768312	rs17697394
0	2	0	2	1	1	0	2	0	2	1	...	0	1	1
1	0	0	2	1	1	0	2	1	2	0	...	0	1	0
2	1	0	2	2	0	0	2	0	2	1	...	0	0	1
3	0	0	1	2	1	0	2	1	2	0	...	0	0	1
4	1	0	2	1	0	0	2	0	2	2	...	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4995	2	0	1	2	0	0	2	1	2	1	...	0	0	0
4996	0	0	1	2	1	0	2	0	2	0	...	0	1	0
4997	0	0	1	2	1	0	2	1	2	1	...	0	1	0
4998	1	0	1	2	1	0	2	0	2	0	...	0	0	1
4999	1	0	0	1	0	0	2	0	0	1	...	0	0	0

5000 rows × 1555 columns

Figure 4: Figure 4. Example of a genotype matrix for a European population.

Ultimately, genotype-phenotype modeling can be described by the following graph:



HapGen2
Generate Data for Populations based on full Genome HapMap3
Sample1
Sample 3
Sample 2
Generate Controls in .gen and . sample format for one population
Generate Controls in .gen and . sample format for another population
One Population
Controls Genotype File
Controls Sample File
Another Population
Controls Genotype File
Controls Sample File
Phenotype Simulator
Generated by HapGen the first population data pass to PhenotypeSimulator
Causal SNPs
Phenotype
Use the same SNPs to Generate Phenotype for the second population
Generated by HapGen the second population data pass to PhenotypeSimulator
Causal SNPs
Phenotype
.bed .bim .fam files with causal SNPs from the first population
.bed .bim .fam files with causal SNPs from the second population
.bed .bim .fam files convert to .vcf
.bed .bim .fam files convert to .vcf
Final genotype-phenotype
Final genotype-phenotype
PLINK --assoc and filtering SNPs by p-value <0.05
PLINK --assoc and filtering SNPs by p-value <0.05
Filtering VCF file by SNPs with p-values < 0.05
Filtering VCF file by SNPs with p-values < 0.05

Figure 5: Figure 5. The process of genotype-phenotype modeling.

5 Research architecture

It is assumed that a model trained on a European population dataset will better predict the phenotype values of another European population dataset. Conversely, if the model is trained on an African population, the phenotype of the European population will be predicted less accurately. In this study, we conducted a series of experiments to assess the influence of population genotype on phenotype prediction. We will also analyze which models predicted the phenotype better than others.

In our study, we used the following machine learning and deep learning models: Logistic Regression, Gradient Boosting, Random Forest, SVM, multilayer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN), ensemble of convolutional and recurrent neural networks (CNN + RNN).

Table 1: Research experiments.

Training set	**European population**	**African population**	**Asian population**
Test set			
European population	MLP, CNN, RNN, CNN+RNN, Logistic Regression, Gradient Boosting, SVM, Random Forest	MLP, CNN, RNN, CNN+RNN, Logistic Regression, Gradient Boosting, SVM, Random Forest	MLP, CNN, RNN, CNN+RNN, Logistic Regression, Gradient Boosting, SVM, Random Forest
African population	MLP, CNN, RNN, CNN+RNN, Logistic Regression, Gradient Boosting, SVM, Random Forest	MLP, CNN, RNN, CNN+RNN, Logistic Regression, Gradient Boosting, SVM, Random Forest	MLP, CNN, RNN, CNN+RNN, Logistic Regression, Gradient Boosting, SVM, Random Forest
Asian population	MLP, CNN, RNN, CNN+RNN, Logistic Regression, Gradient Boosting, SVM, Random Forest	MLP, CNN, RNN, CNN+RNN, Logistic Regression, Gradient Boosting, SVM, Random Forest	MLP, CNN, RNN, CNN+RNN, Logistic Regression, Gradient Boosting, SVM, Random Forest

5.1 MLP Architecture

The multilayer perceptron included 4 linear layers, Dropout, Batch Normalization, and the ReLu activation function. The predicted values were activated using the Sigmoid function. The code for the multilayer perceptron architecture can be found in the appendix.

5.2 CNN Architecture

The convolutional neural network consisted of 2 convolutional layers, 2 linear layers, ReLu activation function, and Dropout. Predicted values were activated using the Sigmoid function. The CNN architecture code can be found in the appendix.

5.3 RNN Architecture

The recurrent neural network consisted of a recurrent layer, a linear layer, and Dropout. The predicted values were activated using the Sigmoid function. The RNN architecture code can be found in the appendix.

5.4 CNN+RNN Architecture

The convolutional neural network consisted of two convolutional layers, two linear layers, ReLu activation function, and Dropout. Predicted values were activated using the Sigmoid function. The recurrent neural network consisted of a recurrent layer, a linear layer, and Dropout. The predicted values were activated using the Sigmoid function. The recurrent neural network was called before the convolutional one. The architecture code can be found in the appendix.

For each model, a selection of hyperparameters was made, the values of which are available in the application.

6 Results

Table 2: Results of machine learning and deep learning algorithms. Training sample: European population; test sample: European population.

	MLP	CNN	RNN	CNN-RNN	Logistic Regression	Gradient Boosting	SVM	Random Forest
ROC-AUC	0.891	0.883	0.877	0.879	0.851	0.776	0.889	0.622
Accuracy	0.807	0.805	0.795	0.792	0.762	0.703	0.812	0.572
F1-score	0.807	0.811	0.785	0.786	0.762	0.703	0.812	0.570
Recall	0.828	0.857	0.765	0.781	0.761	0.703	0.812	0.572
Precision	0.788	0.770	0.806	0.791	0.763	0.704	0.813	0.579

Table 3: Results of machine learning and deep learning algorithms. Training sample: European population; test sample: African population.

	MLP	CNN	RNN	CNN + RNN	Logistic Regression	Gradient Boosting	SVM	Random Forest
ROC-AUC	0.538	0.525	0.479	0.521	0.505	0.492	0.487	0.491
Accuracy	0.529	0.513	0.501	0.525	0.501	0.510	0.507	0.489
F1-score	0.655	0.543	0.540	0.606	0.499	0.510	0.470	0.487
Recall	0.864	0.559	0.565	0.706	0.501	0.510	0.507	0.489
Precision	0.527	0.528	0.516	0.531	0.501	0.510	0.509	0.488

Table 4: Results of machine learning and deep learning algorithms. Training sample: European population; test sample: Asian population.

	MLP	CNN	RNN	CNN + RNN	Logistic Regression	Gradient Boosting	SVM	Random Forest
ROC-AUC	0.505	0.485	0.481	0.483	0.495	0.473	0.491	0.5
Accuracy	0.514	0.491	0.484	0.476	0.494	0.491	0.495	0.496
F1-score	0.492	0.459	0.453	0.425	0.494	0.480	0.49	0.496
Recall	0.466	0.429	0.425	0.385	0.494	0.491	0.495	0.496
Precision	0.52	0.494	0.486	0.475	0.496	0.484	0.491	0.498

6.1 Analysis of machine learning and deep learning models in PRS prediction

When analyzing machine learning and deep learning models for PRS prediction, we relied on experiments where the training and test samples belonged to the same population.

Table 5: Results of machine learning and deep learning algorithms. Training sample: African population; test sample: African population.

	MLP	CNN	RNN	CNN + RNN	Logistic Regression	Gradient Boosting	SVM	Random Forest
ROC-AUC	0.889	0.876	0.873	0.867	0.853	0.794	0.896	0.647
Accuracy	0.771	0.776	0.788	0.775	0.768	0.727	0.804	0.613
F1-score	0.806	0.807	0.792	0.773	0.767	0.727	0.803	0.612
Recall	0.922	0.903	0.781	0.744	0.768	0.727	0.804	0.613
Precision	0.716	0.729	0.803	0.805	0.768	0.727	0.804	0.613

Table 6: Results of machine learning and deep learning algorithms. Training sample: African population; test sample: European population.

	MLP	CNN	RNN	CNN + RNN	Logistic Regression	Gradient Boosting	SVM	Random Forest
ROC-AUC	0.485	0.481	0.482	0.480	0.503	0.509	0.493	0.513
Accuracy	0.486	0.493	0.487	0.497	0.486	0.494	0.488	0.513
F1-score	0.457	0.546	0.514	0.562	0.484	0.492	0.487	0.513
Recall	0.443	0.625	0.556	0.660	0.486	0.494	0.488	0.513
Precision	0.472	0.485	0.478	0.489	0.490	0.499	0.493	0.515

Table 7: Results of machine learning and deep learning algorithms. Training sample: African population; test sample: Asian population.

	MLP	CNN	RNN	CNN + RNN	Logistic Regression	Gradient Boosting	SVM	Random Forest
ROC-AUC	0.519	0.519	0.517	0.512	0.494	0.479	0.513	0.483
Accuracy	0.512	0.508	0.511	0.508	0.493	0.487	0.509	0.484
F1-score	0.561	0.610	0.578	0.550	0.493	0.487	0.509	0.484
Recall	0.619	0.763	0.667	0.597	0.493	0.487	0.509	0.484
Precision	0.513	0.507	0.511	0.510	0.494	0.487	0.509	0.484

Table 8: Results of machine learning and deep learning algorithms. Training sample: Asian population; test sample: Asian population.

	MLP	CNN	RNN	CNN + RNN	Logistic Regression	Gradient Boosting	SVM	Random Forest
ROC-AUC	0.868	0.851	0.851	0.844	0.829	0.773	0.866	0.647
Accuracy	0.736	0.749	0.771	0.762	0.756	0.696	0.78	0.597
F1-score	0.775	0.711	0.767	0.748	0.756	0.696	0.780	0.596
Recall	0.906	0.615	0.748	0.704	0.756	0.696	0.78	0.597
Precision	0.678	0.844	0.787	0.799	0.759	0.697	0.780	0.602

Table 9: Results of machine learning and deep learning algorithms. Training sample: Asian population; test sample: European population.

	MLP	CNN	RNN	CNN + RNN	Logistic Regression	Gradient Boosting	SVM	Random Forest
ROC-AUC	0.504	0.502	0.478	0.502	0.510	0.511	0.512	0.512
Accuracy	0.499	0.513	0.501	0.518	0.508	0.516	0.493	0.512
F1-score	0.550	0.554	0.539	0.567	0.508	0.516	0.456	0.510
Recall	0.627	0.619	0.564	0.646	0.508	0.516	0.493	0.512
Precision	0.490	0.501	0.515	0.5056	0.509	0.516	0.512	0.516

Table 10: Results of machine learning and deep learning algorithms. Training sample: Asian population; test sample: African population.

	MLP	**CNN**	**RNN**	**CNN + RNN**	**Logistic Regression**	**Gradient Boosting**	**SVM**	**Random Forest**
ROC-AUC	0.533	0.537	0.534	0.531	0.508	0.520	0.509	0.476
Accuracy	0.525	0.542	0.532	0.539	0.492	0.511	0.502	0.481
F1-score	0.560	0.580	0.541	0.614	0.484	0.501	0.471	0.480
Recall	0.586	0.613	0.533	0.711	0.492	0.511	0.502	0.481
Precision	0.537	0.551	0.5487	0.541	0.492	0.512	0.502	0.481

Thus, in all three similar experiments, among deep learning models, the multilayer perceptron had the best predictive ability with ROC-AUC values of 0.891, 0.889, and 0.868 in experiments conducted on European, African, and Asian populations independently of each other. Among machine learning models, SVM has the best predictive ability, yielding to MLP in experiments with European and Asian populations. However, in the experiment with the African population, SVM outperforms all deep learning and machine learning models. RNN and CNN architectures are inferior to MLP, but the worst predictive ability among deep learning models is the combination of two architectures, CNN and RNN, nevertheless, the metrics are quite high. Among machine learning models, Random Forest has the worst predictive ability, differing significantly from other models—the maximum ROC-AUC among all populations is 0.647.

6.2 Analysis of the role of population in predicting PRS

The hypothesis put forward at the beginning of the study was confirmed. Models trained on one population have low predictive power and unrepresentative metrics when predicting the phenotype of another population. Thus, the maximum ROC-AUC achieved in a similar experiment is 0.538. Models trained on one population cannot be used to predict the phenotype of other populations. This may be due to the fact that the populations we studied have very strong differences in genotype.

7 Conclusion

Further research may include simulating a larger number of individuals, increasing the number of populations—possibly dividing them into subpopulations, using other simulation approaches, such as HAPNEST[13], including comparisons with existing PRS calculation approaches, and testing the method on real data.

8 Literature

[1] Amadeus, S., Li, Y. R., Keating, B. J. (2014). Trans-ethnic genome-wide association studies: advantages and challenges of mapping in diverse populations. Genome Medicine, 6(9), 91.doi:10.1186/s13073-014-0091-5

[2] Badré, A., Zhang, L., Muchero, W., Reynolds, J.C., Pan, C. (2021). Deep neural network improves the estimation of polygenic risk scores for breast cancer. Journal of Human Genetics, 66, 359-369. doi:10.1038/s10038-020-00832-7.

[3] Duan, S., Zhang, W., Cox, N. J., Dolan, M. E. (2008). FstSNP-HapMap3: a database of SNPs with high population differentiation for HapMap3. Bioinformation, 3(3), 139-141. doi:10.6026/97320630003139.

[4] Ju, D., Hui, D., Hammond, D.A., Wonkam, A., Tishkoff, S.A. (2022). Trans-ethnic genome-wide association studies: Importance of Including Non-European Populations in Large Human Genetic Studies to Enhance Precision Medicine. Annual Review of Biomedical Data Science, 5, 321-339. doi:10.1146/annurev-biodatasci-122220-112550.

[5] Huang S, Ji X, Cho M, Joo J, Moore J. DL-PRS: a novel deep learning approach to polygenic risk scores. Preprint from Research Square. 2021 Apr 28. doi: 10.21203/rs.3.rs-423764/v1.

[6] Li, Y.R. Keating, B.J. (2014). Trans-ethnic genome-wide association studies: advantages and challenges of mapping in diverse populations. Genome Medicine, 6, 91. doi:10.1186/s13073-014-0091-5.

[7] Li N., Stephens M.. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data, Genetics, 2003, vol. 165 (pg. 2213-2233)

[8] Meyer, H. V., Birney, E. (2018). PhenotypeSimulator: A comprehensive framework for simulating multi-trait, multi-locus genotype to phenotype relationships. Bioinformatics, 34(17), 2951-2956. doi:10.1093/bioinformatics/bty197.

[9] Montañez, C. A., Fergus, P., Curbelo Montañez, A., Chalmers, C. (2018). Deep Learning Classification of Polygenic Obesity using Genome Wide Association Study SNPs. arXiv preprint arXiv:1804.03198. doi: 10.48550/arXiv.1804.03198.

[10] Purcell, S., et al. (2007). PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses. The American Journal of Human Genetics, 81(3), 559–575. https://doi.org/10.1086/519795

[11] Ruan, Y., Lin, I.-F., Feng, Y.-C., et al. (2022). Improving polygenic prediction in ancestrally diverse populations. Nature Genetics, 54(5), 573-580. doi:10.1038/s41588-022-00992-2.

[12] Su, S., Marchini, J., Donnelly, P. (2011). HAPGEN2: simulation of multiple disease SNPs. Bioinformatics, 27(16), 2304-2305. doi:10.1093/bioinformatics/btr341.

[13]Wharrie, S., Yang, Z., Raj, V., Monti, R., Gupta, R., Wang, Y. (2023). HAPNEST: efficient, large-scale generation and evaluation of synthetic datasets for genotypes and phenotypes. Bioinformatics, 39(9), btad535. doi: 10.1093/bioinformatics/btad535.

[14] Zhou, X., Chen, Y., Ip, F.C.F., et al. (2023). Deep learning-based polygenic risk analysis for Alzheimer's disease prediction. Nature Communications Medicine, 3(49). doi:10.1038/s43856-023-00269-x.

[15] Zhou, G., Chen, T., Zhao, H. (2023). SDPRX: A statistical method for cross-population prediction of complex traits. The American Journal of Human Genetics, 110(1), 13-22. doi: 10.1016/j.ajhg.2022.11.007