# Evaluating Impact of PRS Uncertainty on Ranking and Clinical Decision Making

Martina Fu

## Introduction

Polygenic risk scores (PRSs) are a common method used to enumerate the additive effects of genetic variants on disease risk. In practice, current PRSs are estimations of the true additive effect, and variation in the construction method results in uncertainty in individual PRS uncertainties. In 2022, Ding et al. presented the range of the uncertainty in individual level PRSs through repeated sampling of the LDPred posterior, and suggested methods to incorporate this uncertainty on individuals. However, the effects of PRS uncertainty on ranking and clinical decision making are not well understood.

We explore the effects of PRS uncertainty on the ranking of individuals based on various criteria here. Specifically, we consider the effects of PRS uncertainty on risk prediction for disease incidence by comparing the efficacy of using various ranking methods to identify cases, including methods on the relative risk and absolute risk scale, as presented below. We consider as our benchmark the number of cases in the individuals with the highest scores.

## Model Set-Up

We begin by presenting the models used for our scores.

### Absolute Risk Model

First, we establish a model for calculating the absolute risk for some binary outcome $D$, such as disease incidence. For this model, we assume that the PRS for individual $i$ is given by $PRS_i = G_i\beta_G$, where $G_i$ is the genotype of the individual, and $\beta_G$ is the weights of the genetic variant.

Now, we wish to calculate the absolute risk, or $P(D = 1|PRS)$. For simplicity, we assume a log-linear model, though this may be extended to additional models of the absolute risk, including logistic and time to event. Assuming the log-linear model, for individual $i$, we have:

$$P(D_i = 1|PRS_i) = \exp\left\{\alpha + \gamma PRS_i\right\}$$

where $\alpha$ is the baseline risk for an individual with a PRS value of 0, and $\gamma$ is the effect size of the PRS on the probability of disease.

### PRS Uncertainty Model

We utilize the same model to account for uncertainty in PRS estimation. Recall that $PRS_i|\beta_G, G_i = G_i'\beta_G$ for an individual $i$. In practice, the PRS for individual $i$ is estimated as $E[P\hat{R}S_i|\hat{\beta_G}, G_i] = G_i'E[\hat{\beta_G}]$, where the $\hat{\beta_G}$ is the posterior distribution of the effect weights for the genetic variants. This results in uncertainty in the PRS, which can be quantified by $Var(P\hat{R}S_i|\hat{\beta_G}, G_i) = G_i'Var(\hat{\beta_G})G_i$.

This allows us to calculate the expected value and variance of the absolute risk as follows.

Expected value:

$$E(Pr(D_i = 1|G_i, \hat{\beta_G})|\hat{\beta_G}) = \exp(\alpha) \left( \exp \left( \gamma \times P\hat{R}S_i + \frac{1}{2}\gamma^2 Var(P\hat{R}S_i) \right) \right)$$

Variance:

$$Var(Pr(D_i = 1|G_i, \hat{\beta_G})|\hat{\beta_G}) = \exp(2\alpha + 2\gamma P\hat{R}S_i) \times (\exp(2\gamma^2 Var(P\hat{R}S_i)) - \exp(\gamma^2 Var(P\hat{R}S_i)))$$

We can use these additional considerations when ranking individuals for determining the best method to identify true cases.

## Methods for Ranking

We now present several methods for incorporating the PRS uncertainty to be used to rank individuals' disease risk.

### Mean Relative Risk

One possibility for ranking individuals is by their mean relative risk, or the mean PRS. This is essentially the most commonly used method for ranking individuals, which ignores the PRS uncertainty and simply uses the posterior mean.

$$E[P\hat{R}S_i|\hat{\beta_G}] = G_i\hat{\beta_G}$$

### Threshold Exceedance of Relative Risk

Ding et al. explored the possibility of using a threshold exceedance type method of ranking [1]. Here, we consider the uncertainty surrounding an individual's PRS and rank the population based on the probability of the individuals' true PRSs falling above some threshold, based on the distribution of the PRss obtained from the posterior distribution of the $\beta_G$.

### Expected Rank of Relative Risk

Another method to account for uncertainty considers the variation of the rankings across posterior samples. Specifically, we rank the individuals by their PRS as calculated by each posterior sample, and take for the score the average of these ranks across the samples.

### Mean Absolute Risk

We can also rank the individuals based on scores on the absolute risk scale. One such score is the mean absolute rank, which can be calculated from the mean and variance of the PRS as showed previously.

### Threshold Exceedence of Absolute Risk

We can derive several other scores on the absolute risk scale. For instance, we can calculate a similar threshold exceedance for the absolute risk, using the probability of the absolute risk falling above some threshold as the score.

### Expected Rank of Absolute Risk

Finally, we can again incorporate the PRS uncertainty through the expected rank, by converting all of the PRSs to the absolute risk scale, then finding the mean rank across all of the posterior samples.

## Calculating PRS Uncertainty

We now provide the code for estimating PRS uncertainty by sampling multiple times from the posterior distribution from LDPred in the same manner as in Ding et al. 2022.

This code was provided by Yi Ding, and modified for use in the cluster for the UK Biobank. Rather than using the HapMap3 variants for the LD Matrix, we instead use a subset of the UK Biobank population, which was already calculated by Yuzheng Dun, another PhD student working with Nilanjan.

```
# This script is adapted from https://github.com/privefl/paper-infer/blob/main/code/
# example-with-provided-LD.R
# please download ld ref info and ld ref matrix as instructed in
# https://privefl.github.io/bigsnpr/articles/LDpred2.html
suppressPackageStartupMessages({
    library(bigsnpr)
    library(readr)
    library(tidyverse)
    library(glue)
    library(bigreadr)
    library(matrixStats)
})

################################################################
### Step 1 loading in summary statistics and setting up data
################################################################
# set up parameters

for(chr_num in c(1:22)){

set.seed(246)

ncores <- 4
n_burn_in <- 1000
n_chain <- 10
n_iter <- 500
report_step <- 5
pheno <- 2

ncase <- 76192
ncontrol <- 63082
ss_file <- paste0('/dcs04/nilanjan/data/mfu/prs_simulation/simulated_betas/bcac_ukbb_P',
                  pheno, 'sumstats.txt')

test_bim_tmp <- '/dcs04/nilanjan/data/mfu/ukbb_bfiles/test_files_small/chrCHROM_test'
# CHROM will be replaced by chromosome number later
output_prefix <- paste0('/dcs04/nilanjan/data/mfu/ss_brca/simulated_betas/brca_ukbb_P',
                        pheno, '_out_', chr_num)

ldr = 3/1000

setwd("/fastscratch/myscratch/mfu/")

# load summary statistics
sumstats <- fread2(ss_file, select = c("chr", "pos", "a0", "a1",
                                       "new_beta", "beta_se", "maf.ss"),
                   col.names = c("chr", "pos", "a0", "a1", "beta", "beta_se", "maf"))
sumstats <- sumstats %>%
    filter(chr!='X', chr!='Y') %>%
    mutate(chr = as.integer(chr)) %>%
    drop_na(chr) %>% filter(chr==chr_num) # select autosome only

sumstats$n_eff <- 4 / (1 / ncase + 1 / ncontrol)



################################################################
### Step 2 train PGS
################################################################

# create ld matrix
tmp <- tempfile(tmpdir = "tmp-data")

  obj.bigSNP <- snp_attach(paste0('/dcs04/nilanjan/data/ydun/PRS_Bridge/ref_ukbb/ldpred_ref/
                                  chr',chr_num,'_1000.rds'))

  G   <- obj.bigSNP$genotypes
  CHR <- obj.bigSNP$map$chromosome
  POS <- obj.bigSNP$map$physical.pos

  map <- obj.bigSNP$map[-(2:3)]
  names(map) <- c("chr", "pos", "a0", "a1")

  sumstats$beta <- as.numeric(sumstats$beta)
  info_snp <- snp_match(sumstats, map, strand_flip = T)
  rownames(info_snp) = info_snp$rsid

  POS2 <- snp_asGeneticPos(CHR, POS, ncores = 2)

  ## indices in info_snp
  ind.chr <- which(info_snp$chr == chr_num)
  df_beta <- info_snp[ind.chr, c("beta", "beta_se", "n_eff")]

  nas <- unique(which(is.na(df_beta$beta)), which(is.na(df_beta$beta_se)))

  ind.chr <- ind.chr[!(ind.chr %in% nas)]

  info_snp_small <- info_snp[-nas,]
  if(length(nas) > 0){df_beta <- df_beta[-nas,]}
  ## indices in G
  ind.chr2 <- info_snp$`_NUM_ID_`[ind.chr]

  corr0 <- snp_cor(G, ind.col = ind.chr2, ncores = 4,
                   infos.pos = POS2[ind.chr2], size = ldr) # default

  corr <- as_SFBM(corr0, tmp, compact = TRUE)

# ldscore to get a starting value
(ldsc <- snp_ldsc2(corr0, df_beta))
```

```
h2_est <- abs(ldsc[["h2"]])

# LDPred2 auto
multi_auto <- snp_ldpred2_auto(corr, df_beta, h2_init = h2_est,
                               vec_p_init = seq_log(1e-4, 0.2, length.out = n_chain),
                               ncores = 5, report_step = report_step, num_iter = n_iter,
                               burn_in = n_burn_in, allow_jump_sign = FALSE,
                               shrink_corr = 0.95)

# rescale post_beta_sample to allelic scale, the default scale of snp_ldpred2_auto output is:
# if sumstats is on allelic scale, then beta is allelic scale, beta_sample is std scale
# we need to rescale the effect size to allelic level
effect_scales <- with(df_beta, sqrt(n_eff * beta_se^2 + beta^2))
for (chain_i in 1:length(multi_auto)){
    multi_auto[[chain_i]]$sample_beta <- sweep(multi_auto[[chain_i]]$sample_beta, 1,
                                               effect_scales, '*')
}


# mini chain quality control
all_h2 <- sapply(multi_auto, function(auto) auto$h2_est)
h2 <- median(all_h2)
keep <- between(all_h2, 0.7 * h2, 1.4 * h2)
all_p <- sapply(multi_auto, function(auto) auto$p_est)
p <- median(all_p[keep])
keep <- keep & between(all_p, 0.5 * p, 2 * p)
post_beta_sample <- do.call( 'cbind', lapply(multi_auto[keep], function(auto) auto$sample_beta))

colnames(post_beta_sample) <- paste0("SAMPLE_", seq(1, ncol(post_beta_sample)))

if(length(nas) > 0){
  post_beta_sample <- bind_cols(
    info_snp_small %>% select(
      chr, pos, a0, a1,
    ),
    as_tibble(as.matrix(post_beta_sample))
  )
} else{
  post_beta_sample <- bind_cols(
    info_snp %>% select(
      chr, pos, a0, a1,
    ),
    as_tibble(as.matrix(post_beta_sample))
  )
}

write_tsv(post_beta_sample,  paste0(output_prefix, ".post_beta_sample.tsv.gz"))

}
```

From here, we use PLINK to calculate the PRSs for each set of betas, thereby providing an estimate to the uncertainty in the PRS.

## Assessing Effects of PRS Uncertainty on Risk Prediction

We now calculate the uncertainty in the PRSs, and discuss several methods we can use to rank the risk of individuals.

### Preparing the Data

Let us assume that the polygenic risk scores are saved in a dataframe, with one column for the ID, and then one column for each PRS. For an example, we provide the code for the PRSs generated by sampling from the LDPred2 posterior for breast cancer. Here, we restrict our analysis to unrelated white European females in the UK Biobank, who were not used for the construction of the LD Matrix. First, we calculate the variance of the PRSs, as well as the standardized PRS. We further add the outcomes for the individuals, to determine the efficacy of ranking by each method.

```r
library(data.table)
library(tidyverse)

dat <- fread("/dcs04/nilanjan/data/mfu/ss_brca/test_small/brca_fem_out.post_prs_sample.tsv.gz")

dat <- dat[order(dat$IID),]

prs_means <- dat %>% select(starts_with("SAMPLE")) %>%
  rowMeans()

prs_ses <- dat %>% select(starts_with("SAMPLE")) %>%
  apply(1, sd)

prs_df <- cbind(dat$IID, prs_means, prs_ses)
colnames(prs_df) <- c("ID", "PRS", "SEs")

load("/dcl01/chatterj/data/ameisner/cPRS/outcomePRS20191016.Rdata")

bc_ukbb <- alldata %>% select("subjectID", "sex", "brca_baseline", "brca_incident",
                              "PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7",
                              "PC8", "PC9", "PC10")

brca_full <- merge(prs_df, bc_ukbb, by.x = "ID", by.y = "subjectID")

brca_full <- brca_full %>% filter(sex == "Female")

ref_pop <- read.table("/dcs04/nilanjan/data/ydun/PRS_Bridge/ref_ukbb/ldpred_ref/1000.txt",
                      header = F)

all_unrelated <- read.table("/dcl01/chatterj/data/ukbiobank/cleaning_information_data/
                            unrelated_european_ancestry_eid.txt", header = T)

brca_full <- brca_full %>% filter(ID %in% all_unrelated$eid)

brca_full <- brca_full %>% filter(!(ID %in% ref_pop$V1))
brca_inc <- brca_full %>% filter(brca_baseline == 0)

brca_inc$stdprs <- brca_inc$PRS / sd(brca_inc$PRS)
```

These values can then be used to calculate various scores for use in ranking risk.

### Mean Relative Risk

As mentioned before, one method for ranking is the relative risk, or just the polygenic risk scores. For this, we use the scaled mean PRS as our score to rank by for the relative risk, as given below:

```
logfit <- glm(brca_incident ~ stdprs + PC1 + PC2 + PC3 +
                PC4 + PC5 + PC6 + PC7 + PC8 + PC9 + PC10,
              family = binomial(link = "logit"), data = brca_inc)
summary(logfit)

brca_inc$scprs <- brca_inc$stdprs * (coefficients(logfit)[2])
brca_inc$scvar <- brca_inc$SEs * (coefficients(logfit)[2])

rank_of_mean_prs <- rank(brca_inc$scprs)
```

This method does not take the uncertainty of the PRS into account, so we further construct scores that incorporate the uncertainty into account in two different manners.

### Threshold Exceedance

One method to incorporate the uncertainty of the PRS into the ranking method is to determine the probability of the PRS of exceeding a threshold (here, the 80th percentile of the scaled PRSs), in a manner similar to that of Ding et al 2022. The code to calculate that is as follows:

```
c <- quantile(brca_inc$scprs, probs = 0.8)
brca_inc$ci_score <- 1 - pnorm(c, mean = brca_inc$scprs, brca_inc$scvar)

rank_of_threshold_exceedence <- rank(brca_inc$ci_score)
```

However, this is just one method of comparing rankings. We consider another method to account for the uncertainty in the relative risk scale.

### Expected Rank of Relative Risk

Another possibility is to compare the expected rankings across each PRS calculated by the different LDPred posterior samples, which we calculate as follows.

```
prs_mat <- dat %>% select(starts_with("SAMPLE"))

expected_rank_of_prs <- prs_mat %>% apply(2, rank) %>% rowMeans()
```

### Mean Absolute Risk

Another method of ranking is based on the theoretical expected absolute risk, which is calculated by $E(Pr(D_i = 1|G_i)) = \exp(\alpha)\left(\exp\left(\beta \times P\hat{R}S_i + \frac{1}{2}\beta^2 Var(P\hat{R}S_i)\right)\right)$.

```
brca_inc$abs_risk <- exp(coefficients(logfit)[1] + brca_inc$scprs + 0.5 * brca_inc$scvar^2)

rank_of_abs_risk <- rank(brca_inc$abs_risk)
```

### Threshold Exceedance of Absolute Risk

We can include the variance of the absolute risk in our score by calculating the threshold exceedance on the absolute risk scale. To do this, we can calculate the empirical probability of the absolute risk falling above some threshold. For this example, we again use the 80th percentile of the absolute risk scores.

```
absfunc <- function(vec){
  return(exp((coefficients(logfit)[1]) + vec * (coefficients(logfit)[2])))
```

```r
}

absrisk_mat2 <- apply(prs_mat, 2, absfunc)

absrisk_mean2 <- absrisk_mat2 %>% rowMeans()

cemp <- quantile(absrisk_mean2, probs = 0.8)

binmat = absrisk_mat2 > cemp

probvec = rowMeans(binmat)

brca_inc$probemp = probvec

expected_rank_of_absrisk = rank(brca_in$probemp)
```

### Expected Rank of Absolute Risk

One last method to use for ranking is the expected rank of the absolute risk, across the samples. For this, we again find the rank across each sample, and then take the mean of the ranks across the samples, as follows:

```r
exp_absrank = apply(absrisk_mat2, 2, rank)

mean_expabsrisk = exp_absrank %>% rowMeans()

expected_rank_of_absolute_risk = rank(absrisk_mean2)
```

### Comparing Efficacy in Identifying Cases

Finally, to ascertain the efficacy of each of these scores to identify cases, we compare the number of cases each score can identify from the top $n$ individuals in the UK Biobank who are disease free at baseline. In this example, we provide the code for the top 1000 individuals.

```r
brca_inc$exprank_prs = expected_rank_of_prs
brca_inc$exprank_absrisk = expected_rank_of_absolute_risk

brca_full <- brca_inc

ordervec <- rep(0, 6)

ordermat <- brca_full[order(-brca_full$scprs)[c(1:1000)],]
ordervec[1] <- sum(ordermat$brca_incident)

ordermat <- brca_full[order(-brca_full$exprank_prs)[c(1:1000)],]
ordervec[2] <- sum(ordermat$brca_incident)

ordermat <- brca_full[order(-brca_full$ci_score)[c(1:1000)],]
ordervec[3] <- sum(ordermat$brca_incident)

ordermat <- brca_full[order(-brca_full$abs_risk)[c(1:1000)],]
ordervec[4] <- sum(ordermat$brca_incident)

ordermat <- brca_full[order(-brca_full$probemp)[c(1:1000)],]
ordervec[5] <- sum(ordermat$brca_incident)
```

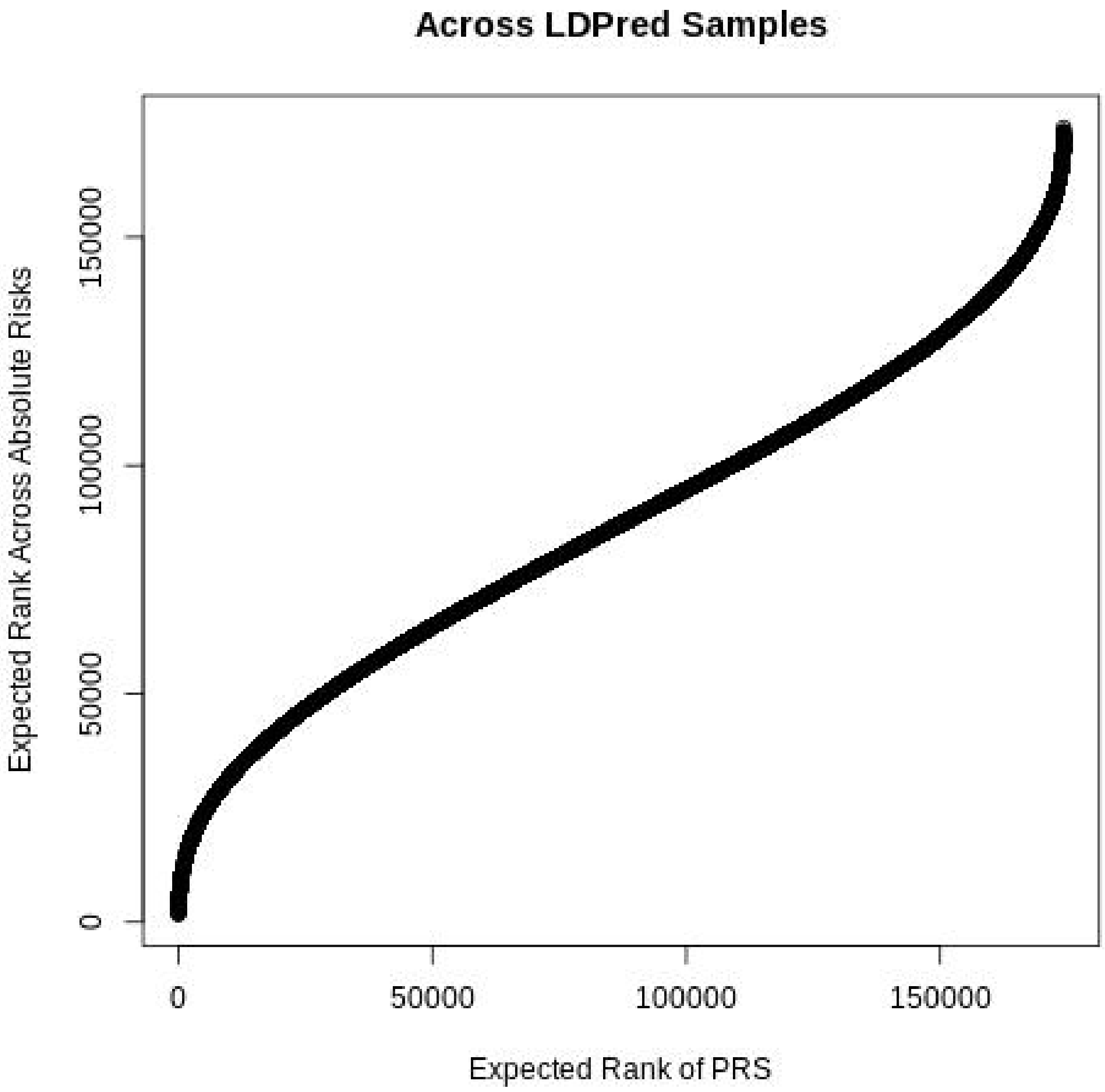
```
ordermat <- brca_full[order(-brca_full$exprank_absrisk)[c(1:1000)],]
ordervec[6] <- sum(ordermat$brca_incident)

ordervec
```

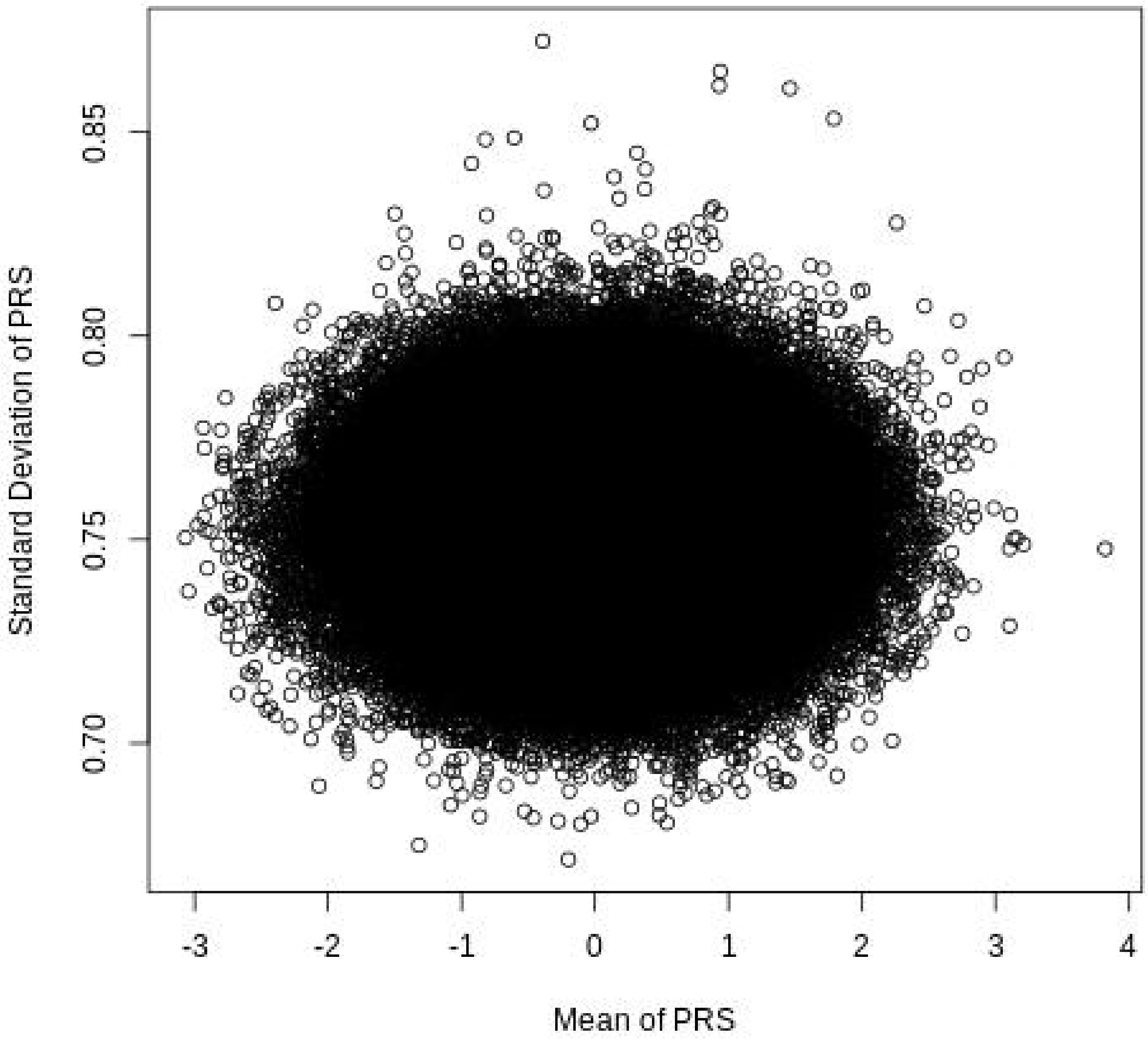
## Sample Output

We provide some examples of the output that can be obtained through the code here. In the examples provided, the PRSs were calculated for breast cancer from the BCAC summary statistics, and analyzed using the methods listed above.

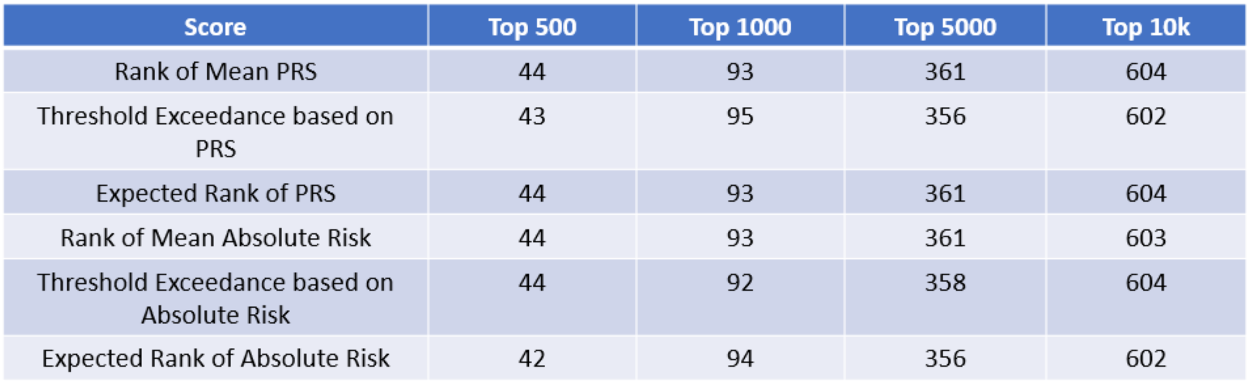
The various methods of ranking individuals may be compared by plotting the ranks against each other. For example, the following plot was obtained by plotting the expected rank of the absolute risk against the expected rank of the PRS.



We can additionally plot the mean and the variance of the PRSs, as shown here.



Finally, we provide an example of the type of table we can obtain from the efficacy results.



| Score | Top 500 | Top 1000 | Top 5000 | Top 10k |
|---|---|---|---|---|
| Rank of Mean PRS | 44 | 93 | 361 | 604 |
| Threshold Exceedance based on PRS | 43 | 95 | 356 | 602 |
| Expected Rank of PRS | 44 | 93 | 361 | 604 |
| Rank of Mean Absolute Risk | 44 | 93 | 361 | 603 |
| Threshold Exceedance based on Absolute Risk | 44 | 92 | 358 | 604 |
| Expected Rank of Absolute Risk | 42 | 94 | 356 | 602 |

## References

[1] Ding, Y., Hou, K., Burch, K.S. et al. Large uncertainty in individual polygenic risk score estimation impacts PRS-based risk stratification. Nat Genet 54, 30–39 (2022). https://doi.org/10.1038/s41588-021-00961-5