Polygenic Risk Score Analyses Workshop 2022

Day 2: Introduction to Polygenic Risk Scores

Day 2 Timetable

Time	Title	Presenter
9:00 - 9:15	Arrival	-
9:15 - 10:30	Lecture: Introduction to PRS I	Dr Paul O'Reilly
10:30 - 11:00	Coffee Break and Q&A	-
11:00 - 12:00	Lecture: Introduction to PRS II	Dr Paul O'Reilly
12:00 - 13:30	Lunch	-
13:30 - 14:30	Practical: Introduction to PRS I	Dr Conrad Iyegbe & Tutors
14:30 - 15:00	Coffee Break and Q&A	-
15:00 - 16:00	Practical: Introduction to PRS II	Dr Conrad Iyegbe & Tutors
16:00 - 17:00	Special Seminar: Polygenic Risk Scores + and Ethics	Dr Nicki Tiffin

Contents

1 Introduction to Polygenic Score Analyses

1.1 Key Learning Outcomes

After completing this practical, you should be able to:

1. Perform basic Polygenic Risk Score (PRS) analyses using PRSice (Euesden, Lewis & O'Reilly 2015; Choi & O'Reilly 2019)
2. Interpret results generated from PRS analyses
3. Customise visualisation of results

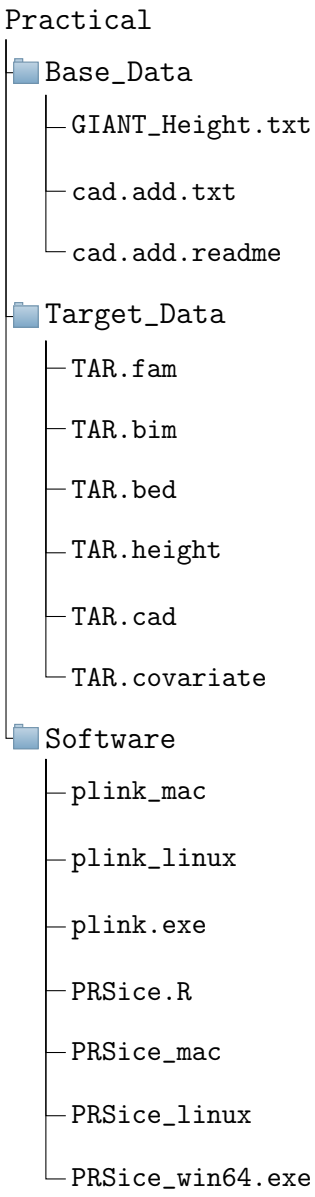
1.2 Resources you will be using

To perform PRS analyses, summary statistics from Genome Wide Association Studies (GWAS) are required. In this workshop, the following summary statistics are used:

Phenotype	Provider	Description	Download Link
Height	GIANT Consortium	GWAS of height on 253,288 individuals (**wood_defining_2014**)	Download Link
Coronary artery disease (CAD)	CARDIoGRAM plus C4D consortium	GWAS on 60,801 CAD cases and 123,504 controls (**consortium_comprehensive_2015**)	Download Link

1.3 Data Structure

You will find all practical materials in the **PRS_Workshop/Day_2** directory. Relevant materials that you should see there at the start of the practical are as follows:



All target phenotype data in this workshop are **simulated**. They have no specific biological meaning and are for demonstration purposes only.

1.4 Introduction

A PRS is a (usually weak) estimate of an individual's genetic propensity to a phenotype, calculated as a sum of their genome-wide genotypes weighted by corresponding genotype effect sizes obtained from GWAS summary statistics. In the next section we will consider what the effect size means and how it is used in computing PRS.

1.5 Understanding GWAS Summary Statistics

When GWAS are performed on a quantitative trait, the effect size is typically given as a beta coefficient (β) from a linear regression with Single Nucleotide Polymorphism (SNP) genotypes as predictor of phenotype. The β coefficient estimates the increase in the phenotype for each copy of the *effect allele*. For example, if the effect allele of a SNP is **G** and the non-effect allele is **A** , then the genotypes **AA**, **AG** and **GG** will be coded as 0, 1 and 2 respectively. In this scenario, the β coefficient reflects how much the phenotype changes for each **G** allele present (NB. The β can be positive or negative - so the 'effect allele' is simply the allele that was coded in the regression, not necessarily the allele with a positive effect).

When a GWAS is performed on a binary trait (e.g. case-control study), the effect size is usually reported as an Odd Ratios (OR). Using the same example, if the OR from the GWAS is 2 with respect to the **G** allele, then the OR of **AG** relative to **AA** is 2, and the OR of **GG** relative to **AA** is 4. So an individual with the **GG** genotype are estimated* to be 4 times more likely to be a case than someone with the **AA** genotype (*an Odds Ratio is itself an estimate of a Risk Ratio, which cannot be calculated from a case/control study)

The relationship between the β coefficient from a logistic regression and the OR is:

$$OR = e^{\beta}$$
$$log_e\,(OR) = \beta$$

While GWAS usually convert from the β to the OR when reporting results, most PRS software convert OR back to β's ($log_e(OR)$) to allow simple addition of $log_e(OR)$'s.

Column names are not standardised across reported GWAS results, thus it is important to check which column is the effect (coded) allele and which is the non-effect allele. For example, in the height GWAS conducted by the GIANT consortium, the effect allele is in the column **Allele1**, while **Allele2** represents the non-effect allele.

Let us open the Height GWAS file (**GIANT_Height.txt**) and inspect the SNPs at the top of the file. If we only consider SNPs *rs4747841* and *rs878177*, what will the 'PRS' of an individual with genotypes **AA** and **TC**, respectively, be? And what about for an individual with **AG** and **CC**, respectively? (Careful - these are not easy to get correct! This shows how careful PRS algorithms/code need to be)

What do these PRS values mean in terms of the height of those individuals?

1.6 Matching the Base and Target Data sets

The first step in PRS calculation is to ensure consistency between the GWAS summary statistic file (*base data*) and the target genotype file (*target data*). Since the base and target data are generated independently, they often relate to different SNPs - and so the first job is to identify the overlapping SNPs across the two data sets and remove non-overlapping SNPs (this is usually done for you by PRS software). If the overlap is low then it would be a good idea to perform imputation on your target data to increase the number of SNPs that overlap between the data sets.

The next, more tricky issue, is that the genotype encoding between the data sets may differ. For example, while the effect allele of a SNP is **T** in the base data, the effect allele in the target might be **G** instead. When this occurs, *allele flipping* should be performed, where the genotype encoding in the target data is reversed so that **TT**, **TG** and **GG** are coded as 2, 1 and 0. Again, this is usually performed automatically by PRS software.

For SNPs that have complementary alleles, e.g. **A|T**, **G|C**, we cannot be certain that the alleles referred to in the target data correspond to those of the base data or whether they are the 'other way around' due to being on the other DNA strand (unless the same genotyping chip was used for all data). These SNPs are known as *ambiguous SNPs*, and while allele frequency information can be used to match the alleles, we remove ambiguous SNPs in PRSice to avoid the possibility of introducing unknown bias.

1.7 Linkage Disequilibrium in PRS Analyses

GWAS are typically performed one-SNP-at-a-time, which, combined with the strong correlation structure across the genome (Linkage Disequilibrium (LD)), makes identifying the independent genetic effects (or their best proxies if these are not genotyped/imputed) challenging. There are two main options for approximating the PRS that would have been generated from full conditional GWAS: 1. SNPs are *clumped* so that the retained SNPs are largely independent of each other, allowing their effects to be summed, assuming additive effects, 2. all SNPs are included and the LD between them is accounted for.

While option 2 is statistically appealing, option 1 has been most adopted in PRS studies so far, most likely due to its simplicity and the similarity of results of methods using the different options to date (**mak_polygenic_2017**). In this workshop we will consider option 1, implemented in PRSice, but if you are interested in how LD can be incorporated as a parameter in PRS calculation then see the LDpred (**vilhjalmsson_modeling_2015**) and lassosum (**mak_polygenic_2017**) papers.

1.7.1 Performing Clumping

Clumping is the procedure where a SNP data set is 'thinned' by removing SNPs across the genome that are correlated (in high LD) with a nearby SNP that has a smaller association P-value.

SNPs are first sorted (i.e. ranked) by their P-values. Then, starting from the most significant SNP (denoted as the *index SNP*), any SNPs in high LD (eg. $r^2 > 0.1$, with r^2 typically calculated from *phased haplotype* data) with the index SNP are removed. To reduce computational burden, only SNPs that are within e.g. 250 kb of the *index SNP* are *clumped*. This process is continued until no *index SNPs* remain.

Use the command below to perform clumping of the Height GWAS data using PLINK(**chang_second_2015**). First, you will have to navigate to the right folder where the data are stored using the terminal. Open the terminal and type the command below at the terminal prompt:

```
cd ~/Desktop/PRS\_Workshop/
```

Next type the following command (NB. See warning below):

```
./Software/plink_linux
--bfile Target_Data/TAR
--clump Base_Data/GIANT_Height.txt
--clump-p1 1
--clump-snp-field MarkerName
--clump-field p
--clump-kb 250
```

```
--clump-r2 0.1
--out Results/Height
```

You can copy & paste code from this document directly to the terminal, but this can cause problems (e.g. when opened by Preview in Mac) and distort the code. Try using Adobe Reader or first copy & pasting to a text editor (eg. notepad) or use the script file provided that contains all the commands.

The command above performs clumping on the height GWAS using LD calculated based on the **TAR** genotype file. SNPs that have $r^2 > 0.1$ within a 250 kb window of the index SNP are removed. This will generate the **Height.clumped** file, which contains the SNPs retained after clumping.

How many SNPs were in the GIANT_Height.txt file before clumping?

How many SNPs remain after clumping?

If we change the r^2 threshold to 0.2, how many SNPs remain? Why are there now more SNPs remaining?

Why is clumping performed for calculation of PRS? (in the standard approach)

1.8 P-Value Thresholding

Deciding which SNPs to include in the calculation of PRS is one of the major challenges in the field. A simple and popular approach is to include SNPs according to their GWAS association P-value. For example, we may choose to include only the genome-wide significant SNPs from the GWAS because those are the SNPs with significant evidence for association. In the next subsection you will compute PRS from GW-significant SNPs only, and then in the subsequent subsection you will generate multiple PRSs using different P-value thresholds.

1.8.1 Height PRS using GW-significant SNPs only

Use the commands below to run PRSice with GIANT Height GWAS as base data and the height phenotype target data. PRSice will calculate Height PRS in the

target data and then perform a regression of the Height PRS against the target individual's true height values. From the **PRS_Workshop/Day_2** directory, run the following command in the terminal:

```
Rscript ./Software/PRSice.R
--prsice Software/PRSice_linux
--base  Base_Data/GIANT_Height.txt
--target Target_Data/TAR
--snp MarkerName
--A1 Allele1
--A2 Allele2
--stat b
--beta
--pvalue p
--pheno Target_Data/TAR.height
--binary-target F
--bar-levels 5e-8
--no-full
--fastscore
--out Results/Height.gws
```

This command takes the Height GWAS summary statistic file (--base), informs PRSice of the column name for the column containing the SNP ID (--snp), the effect allele (--A1), the non-effect allele (--A2), the effect size (--stat) and the P-value (--pvalue). We also inform PRSice that the effect size is a β coefficient (--beta) instead of an OR. The --binary-target F command informs PRSice that the target phenotype is a quantitative trait and thus linear regression should be performed. In addition, we ask PRSice not to perform high-resolution scoring over multiple thresholds (--fastscore), and to compute the PRS using only those SNPs with P-value $< 5\times10^{-8}$.

The default of PRSice is to perform clumping with an r^2 threshold of 0.1 and a window size of 250kb.

To see a full list of command line options available in PRSice type:

```
./Software/PRSice_linux -h
```

Take some time to have a look through some of these user options. By looking at the user options, work out which user option or options were used to ensure that the command above only calculated 1 PRS at the genome-wide significance level of 5e-8?

PRSice performs strand flipping and clumping automatically and generates the **Height.gws.summary** file, together with other output that we will look into later in the practical. The summary file contains the following columns:

1. **Phenotype** - Name of Phenotype.

2. **Set** - Name of Gene Set. Default is *Base*
3. **Threshold** - Best P-value Threshold
4. **PRS.R2** - Variance explained by the PRS
5. **Full.R2** - Variance explained by the full model (including the covariates)
6. **Null.R2** - Variance explained by the covariates (none provided here)
7. **Prevalence** - The population disease prevalence as indicated by the user (not provided here due to testing continuous trait)
8. **Coefficient** - The β coefficient corresponding to the effect estimate of the best-fit PRS on the target trait in the regression. A one unit increase in the PRS increases the outcome by β
9. **Standard.Error** - The standard error of the best-fit PRS β coefficient (see above)
10. **P** - The P-value relating to testing the null hypothesis that the best-fit PRS β coefficient is zero.
11. **Num_SNP** - Number of SNPs included in the best-fit PRS
12. **Empirical-P** - Only provided if permutation is performed. This is the empirical P-value corresponding to the association test of the best-fit PRS - this controls for the over-fitting that occurs when multiple thresholds are tested.

For now, we can ignore most columns and focus on the **PRS.R2** and the **P** column, which provide information on the model fit.

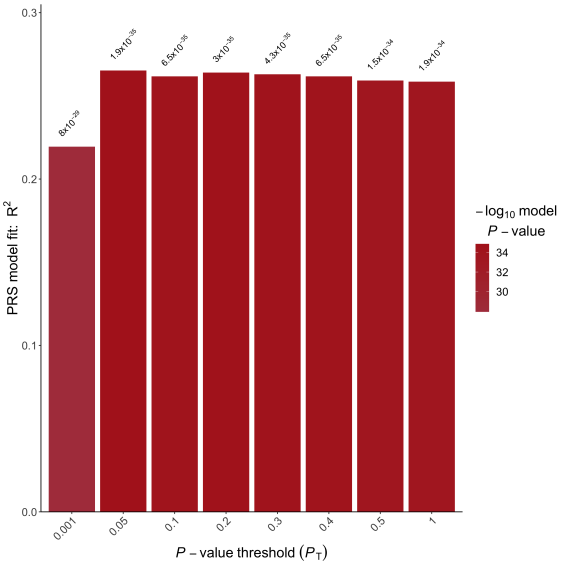
What is the R^2 for the PRS constructed using only genome-wide significant SNPs?

What is the P-value for the association between the PRS and the outcome? Is this significant? (explain your answer).

1.8.2 Height PRS across multiple P-value thresholds

A disadvantage of using only genome-wide significant SNPs is that there are likely to be many true signals among SNPs that did not reach genome-wide significance. However, since we do not know what P-value threshold provides the "best" prediction for our particular data, then we can calculate the PRS under several P-value thresholds and test their prediction accuracy to identify the "best" threshold (NB.

Figure 1.1: BARPLOT generated by PRSice



See **dudbridge_power_2013** for theory on factors affecting the best-fit PRS). This process is implemented in PRSice and can be performed automatically as follows:

```
Rscript ./Software/PRSice.R
--dir .
--prsice Software/PRSice_linux
--base  Base_Data/GIANT_Height.txt
--target Target_Data/TAR
--snp MarkerName
--A1 Allele1
--A2 Allele2
--stat b
--beta
--pvalue p
--pheno Target_Data/TAR.height
--binary-target F
--fastscore
--out Results/Height.fast
```

By removing the --bar-levels and --no-full command, we ask PRSice to perform PRS calculation with a number of predefined thresholds (0.001, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1). The **.prsice** file is very similar to the **.summary** file, the only difference is that **.prsice** file reports the results of model fits for **all thresholds** instead of the most predictive threshold. This allow us to observe the change in model fitting across different thresholds, visualized by the BARPLOT (fig. 1.1) generated by PRSice

Which is the most predictive threshold?

What is the R^2 of the most predictive threshold and how does it compare to PRS generated using only genome-wide significant SNPs?

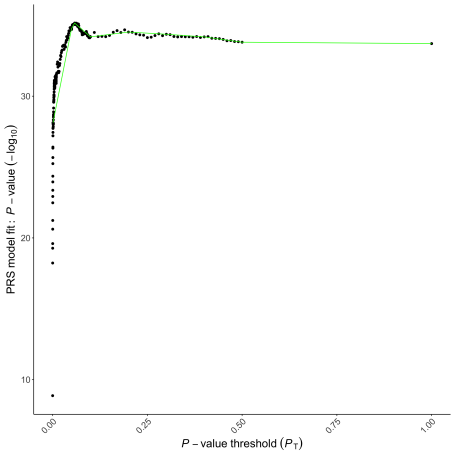
1.8.3 High Resolution Scoring

If we limit ourselves to a small number of P-value thresholds, we might "miss" the most predictive threshold. In order to identify this "best" threshold, we will need "high-resolution scoring", that is, to test the predictive power of PRS generated under a large number of p-value thresholds. We can achieve that by simply removing the --fastscore command from the PRSice script:

```
Rscript ./Software/PRSice.R
--dir .
--prsice Software/PRSice_linux
--base  Base_Data/GIANT_Height.txt
--target Target_Data/TAR
--snp MarkerName
--A1 Allele1
--A2 Allele2
--stat b
--beta
--pvalue p
--pheno Target_Data/TAR.height
--binary-target F
--out Results/Height.highres
```

When PRSice performs high-resolution scoring, it will generate a plot (fig. 1.2) presenting the model fit of PRS calculated at all P-value thresholds.

Figure 1.2: High Resolution Plot generated by PRSice



Which is the most predictive threshold?

How much better is the threshold identified using high-resolution scoring, in terms of model R^2?

How does running fastscore vs high-resolution change the most predictive threshold identified?

The default of PRSice is to iterate the P-value threshold from 5×10^{-8} to 0.5 with a step size of 0.00005, and to include the P-value threshold of 1. Can you identify the commands controlling these parameters?

Accounting for Covariates

When performing PRS, one might want to account for covariates. Based on user inputs, PRSice can automatically incorporate covariates into its model. For example, the following commands will include sex into the regression model as a covariate:

```
Rscript ./Software/PRSice.R
--prsice Software/PRSice_linux
--base  Base_Data/GIANT_Height.txt
--target Target_Data/TAR
--snp MarkerName
--A1 Allele1
--A2 Allele2
```

```
--stat b
--beta
--pvalue p
--pheno Target_Data/TAR.height
--binary-target F
--cov Target_Data/TAR.covariate
--cov-col Sex
--out Results/Height.sex
```

When covariates were include in the analysis, PRSice will use the model fit of **only PRS** for all its output. This $PRS.R^2$ is calculated by minusing the R^2 of the null model (e.g. $Height \sim Sex$) from the R^2 of the full model (e.g. $Height \sim Sex + PRS$).

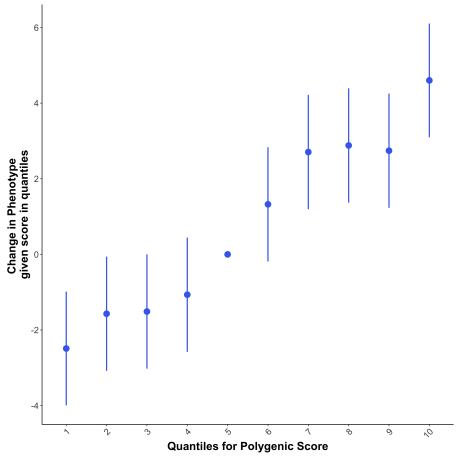
Usually, Principal Components (PCs) are also included as covariates in the analysis to account for population structure and it can be tedious to type all 20 or 40 PCs (e.g. PC1,PC2,...,PC20). In PRSice, you can add **@** in front of the --cov-col string to activate the automatic substituation of numbers. If **@** is found in front of the --cov-col string, any numbers within **[** and **]** will be parsed. E.g. @PC[1-3] will be read as *PC1,PC2,PC3*. Discontinuous input are also supported: @cov[1.3-5] will be parsed as *cov1,cov3,cov4,cov5*. You can also mix it up E.g. @PC[1-3],Sex will be interpret as *PC1,PC2,PC3,Sex* by PRSice.

How does the inclusion of sex as a covariate change the results?

Usually, with categorical variables, dummy variables have to be generated to represent the different categories. Alternatively, residualized phenotype, generated by regresing the covariates against the phenotype, can be used for downstream analyses. A useful feature of PRSice is to automatically generate the dummy variable for users. This can be achieved with the following command:

```
Rscript.exe ./Software/PRSice.R
--prsice Software/PRSice_linux
--base  Base_Data/GIANT_Height.txt
--target Target_Data/TAR
--snp MarkerName
--A1 Allele1
--A2 Allele2
--stat b
--beta
--pvalue p
--pheno Target_Data/TAR.height
--binary-target F
--cov Target_Data/TAR.covariate
--cov-col Sex
--cov-factor Sex
--out Results/Height.sex
```

Figure 1.3: Example of a quantile plot generated by PRSice



1.9 Stratifying Samples by PRS

An interesting application of PRS is to test whether samples with higher PRS have higher phenotypic values. This can be nicely visualized using the quantile plot (fig. 1.3).

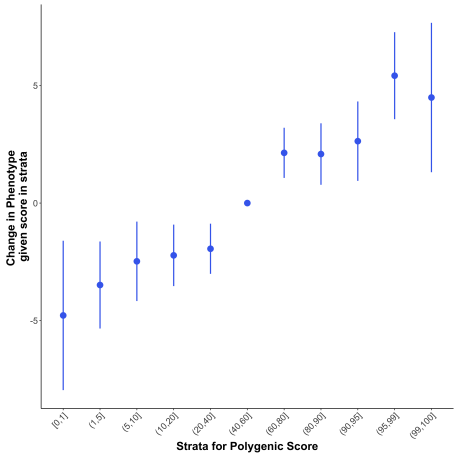
To generate quantile plots in PRSice, simply add --quantile 10 option.

We can skip the PRS calculation using the --plot option, which will use previously calculated PRS to generate the plots

```
Rscript ./Software/PRSice.R
--prsice Software/PRSice_linux
--base  Base_Data/GIANT_Height.txt
--target Target_Data/TAR
--snp MarkerName
--A1 Allele1
--A2 Allele2
--stat b
--beta
--pvalue p
--pheno Target_Data/TAR.height
--binary-target F
--cov Target_Data/TAR.covariate
--cov-col Sex
--plot
--quantile 10
--out Results/Height.sex
```

The --plot option tells PRSice to generate the plots without re-running thw whole PRSice analysis. This is handy when you want to change some of the parameters for plotting e.g. the number of quantiles. **Try running the previous command with 20 quantiles - and again with 50 quantiles - checking the quantile**

Figure 1.4: Example of a strata plot generated by PRSice



plot each time.

A disadvantage of the quantile plot is that it only seperate samples into quantiles of equal size. However, it is somtimes interesting to investigate whether a specific strata (e.g. top 5% of samples), contain a higher PRS than the reference strata. For example, **mavaddat_prediction_2015** found that samples in the highest 1% of PRS distribution have a 2.81 increased OR of breast cancer when comparing to samples at the middle quantiles (40^{th} to 60^{th} percentile). We can mimic their table by using --quant-break, which represents the upper bound of each strata, and --quant-ref, which represents the upper bound of the reference quantile:

```
Rscript ./Software/PRSice.R
--prsice Software/PRSice_linux
--base  Base_Data/GIANT_Height.txt
--target Target_Data/TAR
--snp MarkerName
--A1 Allele1
--A2 Allele2
--stat b
--beta
--pvalue p
--pheno Target_Data/TAR.height
--binary-target F
--cov Target_Data/TAR.covariate
--cov-col Sex
--plot
--quantile 100
--quant-break 1,5,10,20,40,60,80,90,95,99,100
--quant-ref 60
--out Results/Height.sex
```

See the quantile results in Table form in the *_QUANTILES_* file, and the plots in the *_QUANTILES_PLOT_* file. Due to the small sample size of the target data the results here are underwhelming, but with high power we may observe

strong deviations in the extreme quantiles.

1.10 Case Control Studies

In the previous exercises, we have performed PRS analyses on height, which is a quantitative trait. For binary phenotypes (e.g case-control) there are a number of differences in the analysis:

1. Logistic regression has to be performed instead of linear regression
2. ORs are usually provided and need to be converted to β's when constructing PRS

Here we will use CAD as an example. You will find the summary statistic under *Base_Data* (**cad.add.txt**) and the phenotype file (**TAR.cad**) under *Target_Data*. You will also need to specify --binary-target T in the PRSice command to indicate that the phenotype is binary.

GWAS summary statistics for binary traits tend to report the OR instead of the β coefficient, in which case the --or should be used. However, CARDIoGRAM plus C4D consortium provided the β coefficient, therefore we will include --beta in our code and specify --binary-target T to indicate that the phenotype is binary.

```
Rscript ./Software/PRSice.R
--prsice Software/PRSice_linux
--base  Base_Data/cad.add.txt
--target Target_Data/TAR
--snp markername
--A1 effect_allele
--A2 noneffect_allele
--chr chr
--bp bp_hg19
--stat beta
--beta
--pvalue p_dgc
--pheno Target_Data/CAD.pheno
--binary-target T
--out Results/CAD.highres
```

--chr and --bp inform PRSice the columns containing the chromosomal coordinates. This enables PRSice to check whether the SNPs in the Base and Target data have the same chromosomal coordinate.

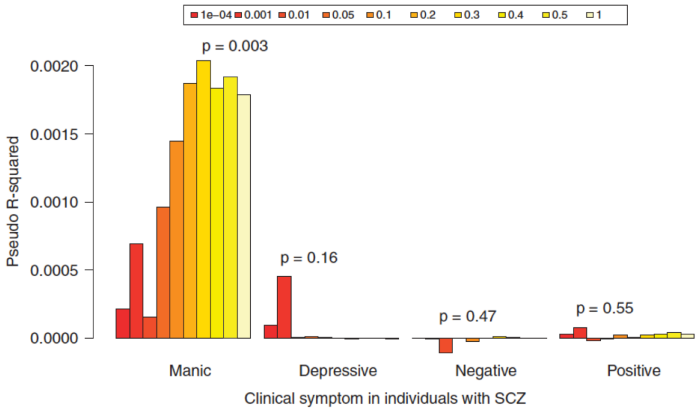


Figure 1.5: Plot taken from Ruderfer et al. 2014

What is the R^2 and P-value of the best-fit PRS?

Does this suggest that there is a significant association between the CAD PRS and CAD status in the target sample?

1.11 Cross-Trait Analysis

A popular application of PRS is in performing cross-trait analyses. This allows some interesting analyses such as those performed by **ruderfer_polygenic_2014** (fig. 1.5), which used the bipolar PRS to predict into different clinical dimensions of schizophrenia.

In this practical, we will perform cross-trait analyses between CAD and Height, using height as the base and CAD as the target.

We will only focus on the simplest form of cross-trait analysis in this practical. To perform the multi-phenotype cross-trait analysis similar to that of **ruderfer_polygenic_2014**, you can use the --pheno-col to include multiple target phenotype into the analysis.

```
1 Rscript ./Software/PRSice.R
2 --prsice Software/PRSice_linux
3 --base Base_Data/GIANT_Height.txt
4 --target Target_Data/TAR
5 --snp MarkerName
6 --A1 Allele1
7 --A2 Allele2
```

```
8  --stat b
9  --beta
10 --pvalue p
11 --pheno Target_Data/CAD.pheno
12 --binary-target T
13 --out Results/Cross.highres
```

What is the R^2 for the most predictive threshold when using height as the base phenotype and CAD as the target phenotype?

Now try using CAD as the base to predict height as the target trait? What is the PRS R^2 for that?