PRSice: Polygenic Risk Score software - Vignette

Jack Euesden, Paul O'Reilly

March 22, 2016

1 The Polygenic Risk Score process

PRSice ('precise') implements a pipeline that has become standard in Polygenic Risk Score (PRS) analyses in recent years, which can be summarised as follows:

- GWAS results are obtained on a phenotype of interest (we call the *base phenotype*)
- A *target data* set is obtained, which contains genotype and phenotype data (the *target phenotype* may be the same or different to the *base phenotype*).
- SNPs present in only one data set are removed, as are 'ambiguous' (A/T or C/G) SNPs and those in linkage disequilibrium with the local SNP with the smallest GWAS *P*-value (the latter is known as *clumping* and can be disabled in PRSice to include all SNPs).
- Polygenic Risk Scores (PRS) are calculated for individuals in the target data as a sum of 'risk alleles' across SNPs with (GWAS) *P*-values below a given *P*-value threshold, weighted by the effect sizes estimated by the GWAS (the scores thus correspond to a *genetic risk/burden* of the base phenotype for each individual).
- A regression is performed to test the association between the PRS and the target phenotype (eg. to test for shared genetic aetiology). Covariates, as well as ancestry informative variables for controlling for population structure (which can be calculated in PRSice), can be included in this regression.
- PRS and the subsequent regressions are calculated at a number of *P*-value thresholds and the model fit of the regressions assessed. PRSice repeats the analysis at 1000s of thresholds in order to identify the most predictive threshold and model.
- Model fit and significance illustrated via bar plots, scatter plots and quantile plots.

PRSice implements this pipeline in a single command via PLINK-1.9, extensive bash scripting for data management, and R to perform the regressions and produce plots - combined to optimise the efficiency of the multi-stage PRS process.

PRSice offers a flexible set of user-options in addition to this standardised pipeline, as well as incorporating the *gtx* R package for which summary statistics are used for both the base and target data.

2 Getting set up to use PRSice

Once you have downloaded PRSice ensure that the .zip file has opened correctly and that you have a directory containing the PRSice R script file, PLINK executable files and TOY example files. **Check that the permissions** are open for you to use the PLINK executable (especially if you have moved it to a cluster) and if not then use the *chmod* command to open them (eg. chmod uga+wrx plink_1.9_linux_160914).

2.1 Setting up R for PRSice

Before you can run PRSice, the packages that it uses must be installed. First, **ensure that you have the latest version of R downloaded - if not then update R!** Next, run the following commands in R:

```
R

> library(fmsb)
> library(batch)
> library(gtx)
Loading required package: survival
Loading required package: splines
> library(plyr)
> library(ggplot2)
>
```

If any of these commands generate errors, it means that the corresponding packages have not previously been downloaded and so users must **install these packages** and their dependancies, using the `install.packages()` function in R.

2.2 Running PRSice from a Windows machine

You can run PRSice by logging on to a Unix/Linux based cluster. To do this you can use a ssh shell, such as mobaXterm, available here. Your IT dept. should be able to get you set-up on a cluster if you are not already.

You will need to copy files needed to run PRSice on to a clusterusing the `scp` command:

```
scp ~/Downloads/PRSice/* user@cluster:/user/home/
```

Where user and cluster are your username and cluster name respectively. Once you have copied files across, you can log into the cluster using the `ssh` command:

```
ssh user@cluster -X
```

NB: You will need to add the '-X' flag to ensure that you can see the plots that are produced.

After this, the final step is to check that you have permission to execute the executable files copied across. This can be done using the `chmod` command:

```
chmod uga+wrx ./*
```

If you cannot log on to a cluster then consider using cygwin or booting Linux on a flash stick.

2.3 Input data files

In the examples that follow we will use the TOY data included in the PRSice directory. The base summary data are in the file `TOY_GWAS.assoc` and the target genotype data are stored as three files in PLINK binary format: `TOY_TARGET.bim, TOY_TARGET.bed` and `TOY_TARGET.fam`. An additional file containing phenotype data for individuals in the target data set, measured on a quantitative trait, are in the file `TOY_TARGET_QUANTITATIVE.pheno`.

The first few lines of the base data can be viewed in terminal using the command `head TOY_BASE_GWAS.assoc`, and are as follows:

```
SNP        CHR  BP         A1  A2  P        OR
SNP_22857  4    103593179  1   2   0.2852   13.29
SNP_13879  2    237416793  1   2   0.8784   21.624
SNP_20771  4    16957461   1   2   0.1994   91.265
SNP_13787  2    235355721  1   2   0.7234   3.178
SNP_25383  4    189927377  1   2   0.3309   3.167
SNP_25290  4    187995996  1   2   0.6327   0.427
SNP_21478  4    40161304   1   2   0.06454  5.066
SNP_12129  2    176643771  1   2   0.9378   1.276
SNP_22809  4    101441465  1   2   0.8111   0.004
```

Take note of the column names. `SNP, A1, P` and an effect size column - either `BETA` or `OR` (Odds Ratio) - are absolutely necessary for PRSice to run. Common errors are due to mislabeling these columns (e.g. BETA instead of OR), or problems with their content (e.g. different naming systems used for SNPs in the base and target data sets, e.g. rs IDs in one and CHR:BP in the other). **The order of the columns does not matter, but the names must be correct.**

The format of the phenotype data is also important. By default, phenotype will be read from column 6 of the fam file of the target data, which can be viewed using `head TOY_TARGET_DATA.fam`.

```
CAS_1   CAS_1   0  0  2  2
CAS_2   CAS_2   0  0  1  2
CAS_3   CAS_3   0  0  1  2
CAS_4   CAS_4   0  0  2  2
CAS_5   CAS_5   0  0  2  2
CAS_6   CAS_6   0  0  2  2
CAS_7   CAS_7   0  0  2  2
CAS_8   CAS_8   0  0  1  2
CAS_9   CAS_9   0  0  2  2
CAS_10  CAS_10  0  0  2  2
```

Case/Control data are coded 1 for control and 2 for cases, with missing values coded 0, -9 or NA. They can also be coded 0 for control, 1 for cases, and -9 or NA for missing. The format of quantitative data can be illustrated by viewing the first few lines of the quantitative phenotype file, `TOY_TARGET_QUANTITATIVE.pheno`:

```
CAS_1 2.11238543474405
CAS_2 2.1026870272198
CAS_3 1.83291829136584
CAS_4 2.0230388140955
CAS_5 2.16146693576679
CAS_6 2.06415915226195
CAS_7 2.28997813368432
CAS_8 2.45419652735488
CAS_9 2.15746025136836
CAS_10 1.63613221166589
```

There are a some important points to note here. The file has no column names. The first column contains the individual IDs - i.e. the *second* column of the .fam file. **Any missing values should be coded as NA**. NB, -9 is accepted as a code for missing data when phenotype is binary, but when a phenotype is quantitative, only NA is accepted.

3 Running PRSice

Here we show 2 examples of running PRSice over the TOY data, illustrating the different figures produced by PRSice. We will disable the modelling of covariates as these are simulated data where other risk factors or population structure were not simulated, and disable clumping as the SNPs have been simulated under linkage equilibrium. We will also specify a relatively small number of thresholds, in order to speed up running time, by choosing a large value for `sinc` - the increments between thresholds, P_T. Increments are calculated between a lower bound of `slower` and an upper bound of `supper`.

3.1 Example 1

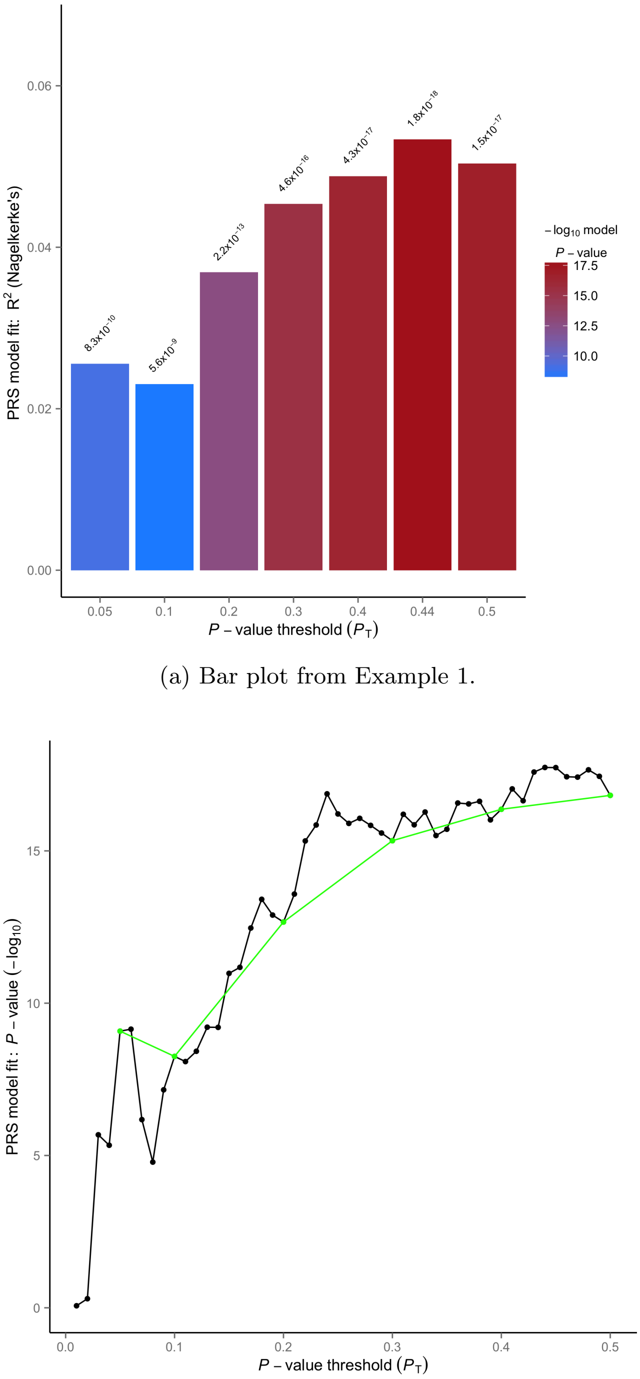
To perform a simple run of PRSice try the following command (use the PLINK mac file if on a Mac):

```
R -q --file=./PRSice_v1.25.R --args \
  base TOY_BASE_GWAS.assoc \
  target TOY_TARGET_DATA \
  slower 0 \
  supper 0.5 \
  sinc 0.01 \
  covary F \
  clump.snps F \
  plink ./plink_1.9_linux_160914 \
  figname EXAMPLE_1
```

After a few minutes, a number of files are generated. Text files, which can be used for further analysis are described fully in the PRSice manual. The barplot which is generated allows users to quickly inspect the model fit for polygenic risk score on target phenotype. Inspecting this bar plot, below, reveals evidence for shared genetic architecture between the two traits (Figure 1a). These results can be seen in more detail by inspecting the high resolution plot, which can be seen below in Figure 1b

There are a few crucial things to note here:

- If plink is in the current working directory, the path to it must begin `./`
- The backslashes here are used to prevent an end of line. Users must not put spaces after these
- Covariates are generated and used by default. This is disabled here using `covary F`



(a) Bar plot from Example 1.

(b) High resolution plot from Example 1.

Figure 1: : Figures generated by PRSice on a simulated binary trait, from Example 1.

3.2 Example 2

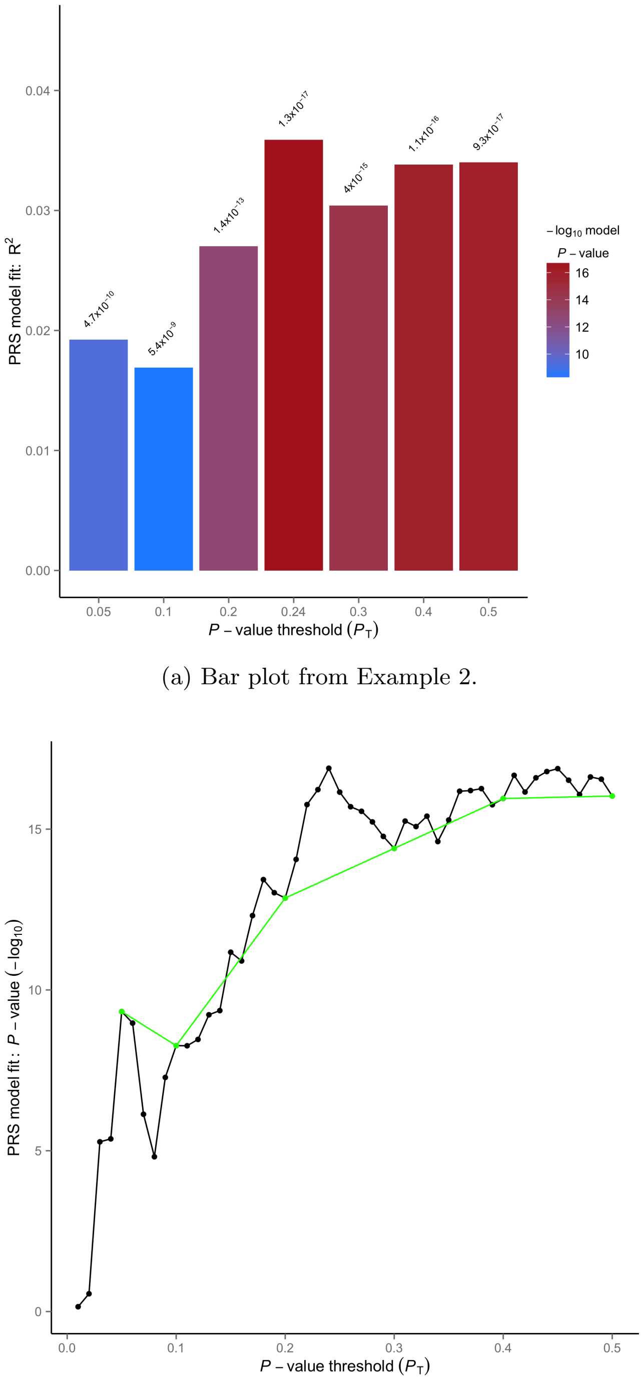
Secondly, we can investigate a quantitative trait in the target data using the external file - `TOY_TARGET_QUANTITATIVE.pheno` - described above. We can run PRSice as above using this command:

```
R -q --file=./PRSice_v1.25.R --args \
  base TOY_BASE_GWAS.assoc \
  target TOY_TARGET_DATA \
  slower 0 \
  supper 0.5 \
  sinc 0.01 \
  covary F \
  clump.snps F \
  plink ./plink_1.9_linux_160914 \
  figname EXAMPLE_2 \
  pheno.file TOY_TARGET_QUANTITATIVE.pheno\
  binary.target F
```

This also reveals evidence for shared genetic overlap between the traits measured in the base and target dataset, as can be seen in the barplot generated (Figure 2a) and the high-resolution plot (Figure 2b).

A few crucial things to note here are:

- Users need to specify if the phenotype used is not dichotomous - this is done here using the option `binary.target F`
- Here a new figure name is set using `figname` in order to prevent PRSice overwriting the previous outputs. If this is not done, PRSice will overwrite previous outputs.



(a) Bar plot from Example 2.

(b) High resolution plot from Example 2.

Figure 2: : Figures generated by PRSice on a simulated quantitative trait, from Example 2.

3.3 Example 3

Quantile plots can be generated to visualise how polygenic score influences risk of disease compared to some reference group. We generate quantile plots using the same data as Example 1 now:

The option to generate quantile plots is enabled using `quantiles T`:

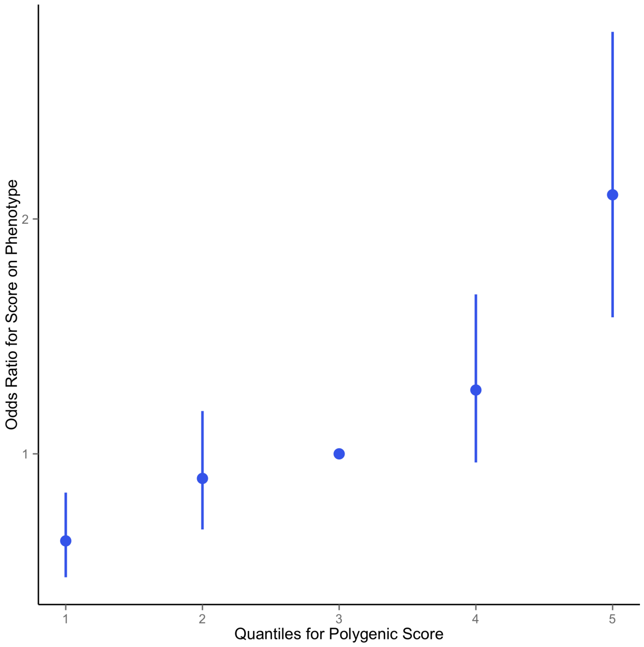
```
R -q --file=./PRSice_v1.25.R --args \
  base TOY_BASE_GWAS.assoc \
  target TOY_TARGET_DATA \
  slower 0 \
  supper 0.5 \
  sinc 0.01 \
  covary F \
  clump.snps F \
  plink ./plink_1.9_linux_160914 \
  figname EXAMPLE_3_QUANTILES_1 \
  quantiles T
```

The graph produced (Figure 3a) shows the Odds of phenotype given an individual is in a particular quantile for polygenic risk score (versus the reference quantile). If target phenotype is quantitative, beta rather than Odds Ratio is used. If `covary T`, these betas or Odds Ratios are adjusted for these covariates.

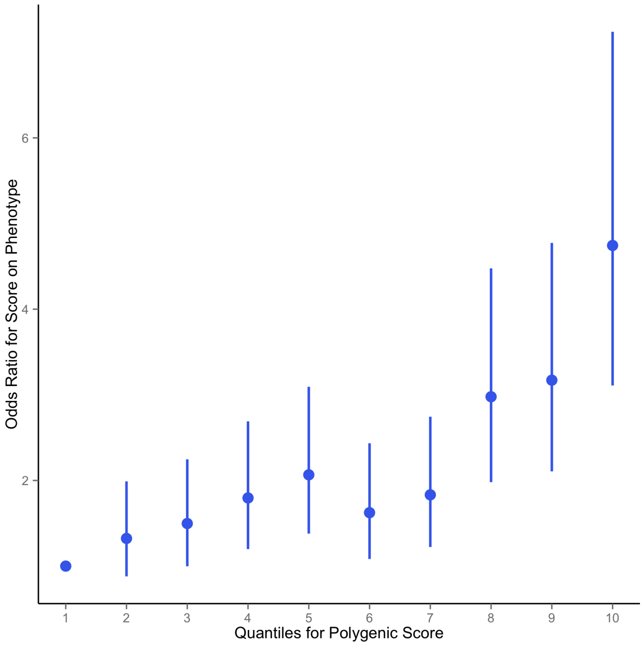
Five quantiles are used by default - larger target data sets or phenotypes with larger effect sizes allow users to use more quantiles. We use the median quantile as a default, in order to show the effect of deviating from a 'population average' level of genetic risk on pedicted phenotype. We now discuss how to change these two options - number of quantiles and reference quantile - using the options `num.quantiles` and `quant.ref` respectively.

```
R -q --file=./PRSice_v1.25.R --args \
  base TOY_BASE_GWAS.assoc \
  target TOY_TARGET_DATA \
  slower 0 \
  supper 0.5 \
  sinc 0.01 \
  covary F \
  clump.snps F \
  plink ./plink_1.9_linux_160914 \
  figname EXAMPLE_3_QUANTILES_2 \
  quantiles T \
  quant.ref 1 \
  num.quantiles 10
```

Here, the first quantile is used as a reference quantile, and the data is binned into 10 quantiles - ‘deciles’ (Figure 3b).



(a) A ‘default’ Quantiles plot generated on data from Example 1.



(b) A Quantiles plot generated on data from Example 1, with default options changed.

Figure 3: : Quantiles plots generated by PRSice on binary target data, using default options and user-specified options.