AMYPAD
Amyloid imaging to prevent
Alzheimer´s Disease

Polygenic Risk Score Computation with PRSice

Emma S. Luckett, PhD
Amsterdam UMC

25.07.2025

www.amypad.eu

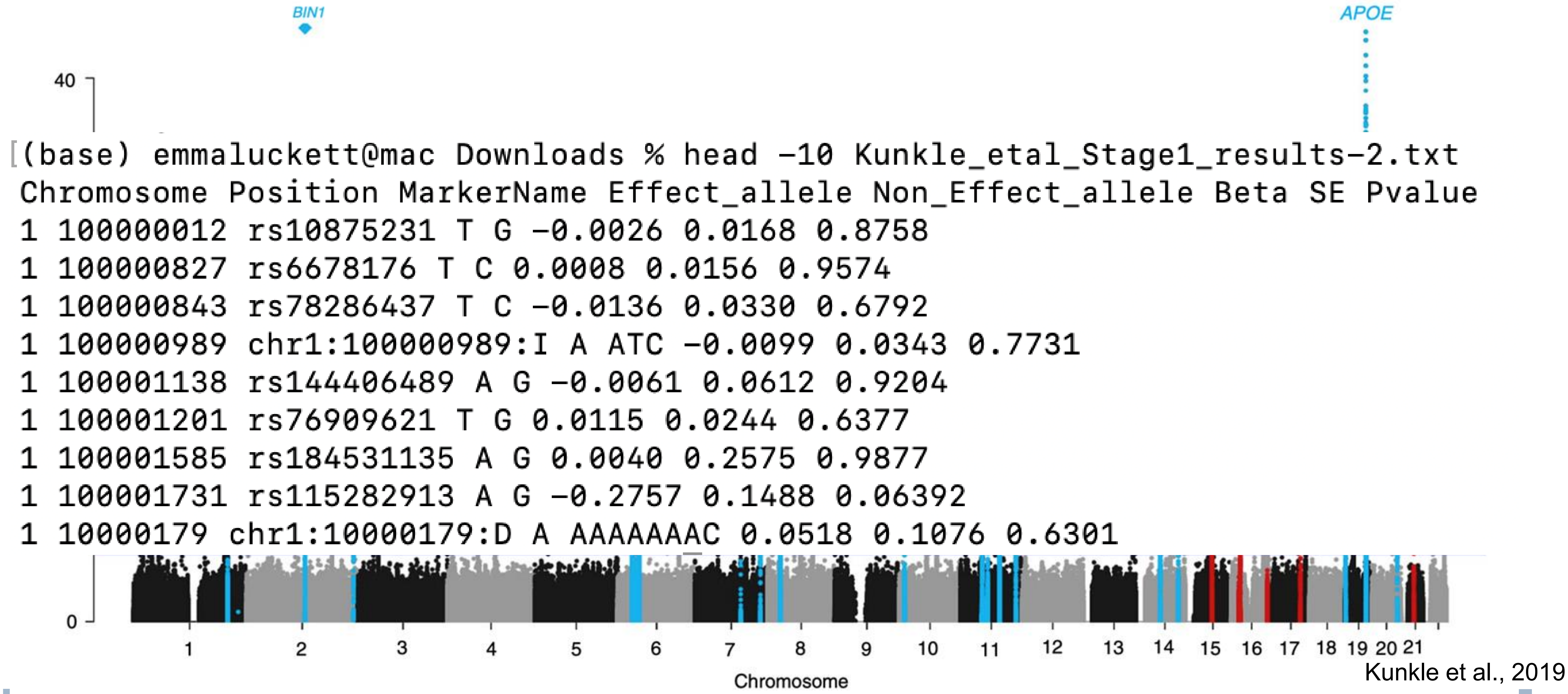
Presentation Overview

1. Recap
2. What is a polygenic risk score and the limitations
3. Why compute a polygenic risk score
4. Summary of tools
5. PRSice workflow, summary statistics, and input data
6. PRSice output
7. ADNI example

Recap

- So far, you've learnt:
  - How to design a genetics experiment
  - How to obtain (genetic) data and perform quality control
  - SNP analysis
  - How to conduct a GWAS

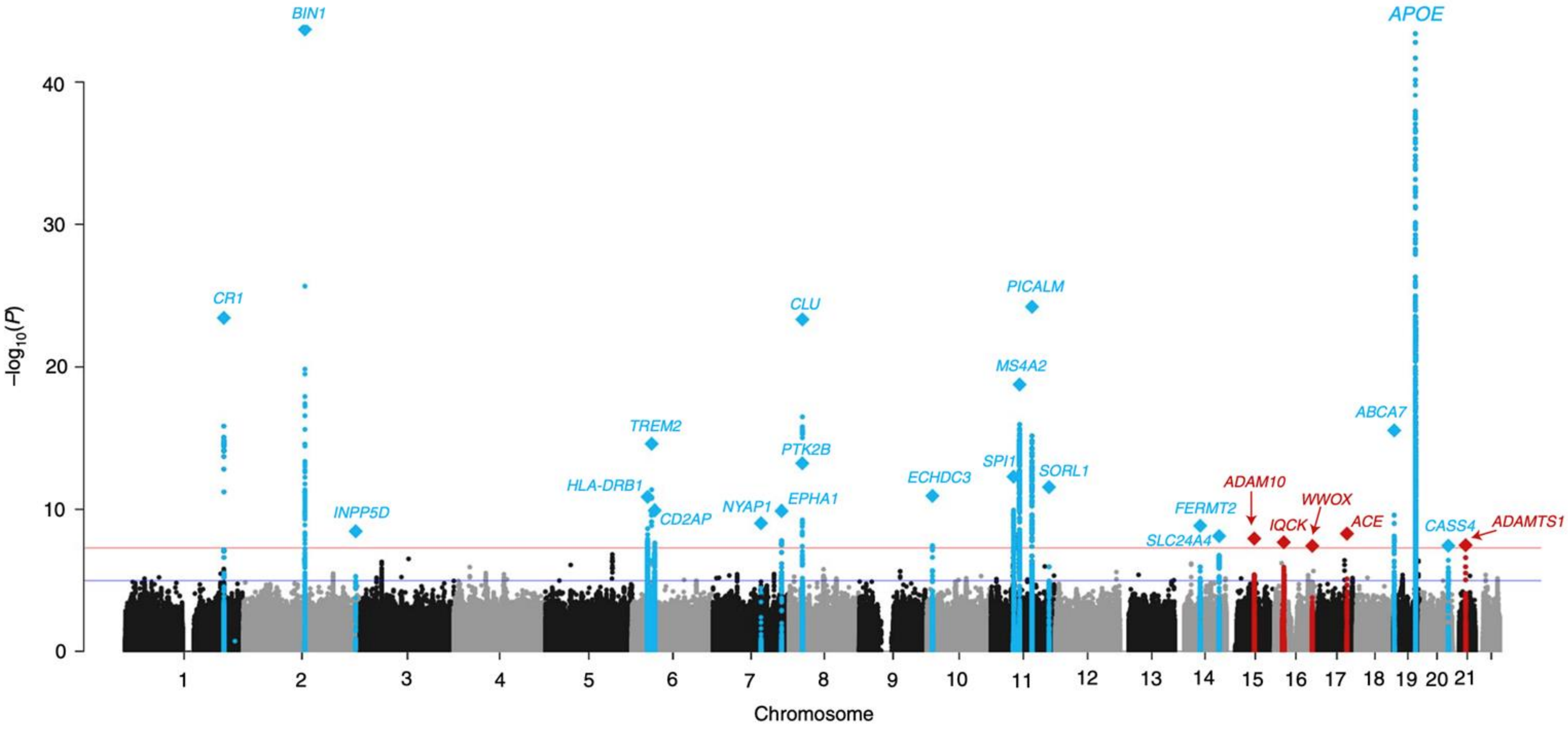
Recap



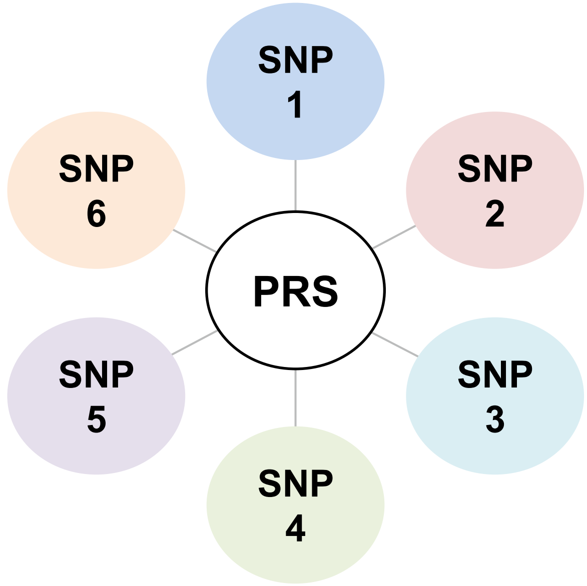
```
[(base) emmaluckett@mac Downloads % head -10 Kunkle_etal_Stage1_results-2.txt
Chromosome Position MarkerName Effect_allele Non_Effect_allele Beta SE Pvalue
1 100000012 rs10875231 T G -0.0026 0.0168 0.8758
1 100000827 rs6678176 T C 0.0008 0.0156 0.9574
1 100000843 rs78286437 T C -0.0136 0.0330 0.6792
1 100000989 chr1:100000989:I A ATC -0.0099 0.0343 0.7731
1 100001138 rs144406489 A G -0.0061 0.0612 0.9204
1 100001201 rs76909621 T G 0.0115 0.0244 0.6377
1 100001585 rs184531135 A G 0.0040 0.2575 0.9877
1 100001731 rs115282913 A G -0.2757 0.1488 0.06392
1 10000179 chr1:10000179:D A AAAAAAAC 0.0518 0.1076 0.6301
```

Kunkle et al., 2019

What is a Polygenic Risk Score (PRS)?



Kunkle et al., 2019



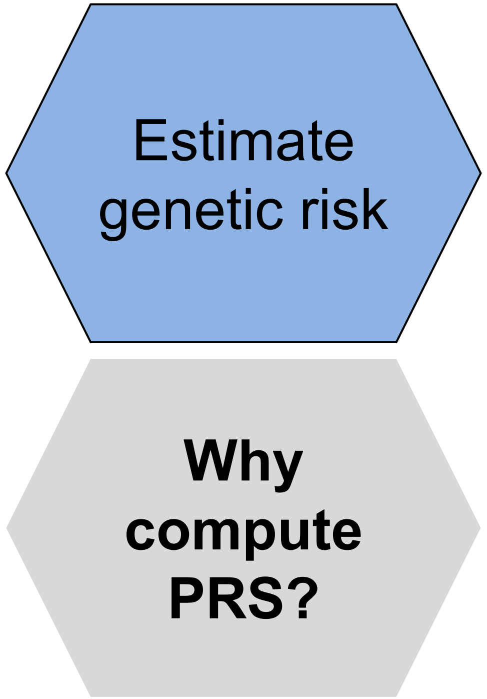
Formula:
PRS = Σ (effect size × genotype)

- Modest predictive power
- Limited cross-ancestry generalisation
- Ignores environmental and epigenetic effects

Why use PRS?

Why compute PRS?

Why use PRS?

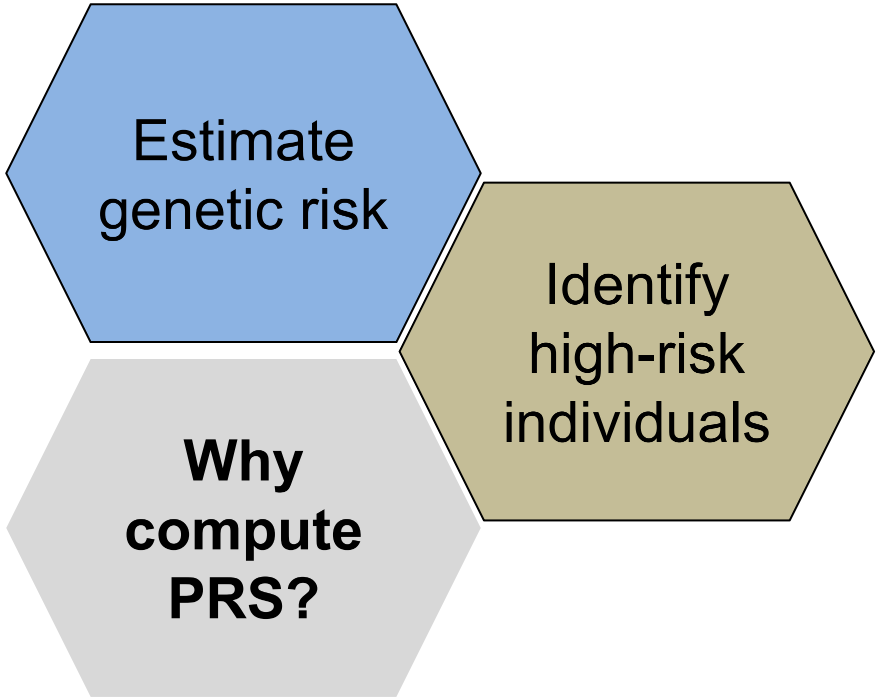


Estimate genetic risk

Why compute PRS?

- PRS aggregates the effects of thousands of variants (SNPs) into a single number that reflects an individual’s risk for a trait or disease

Why use PRS?



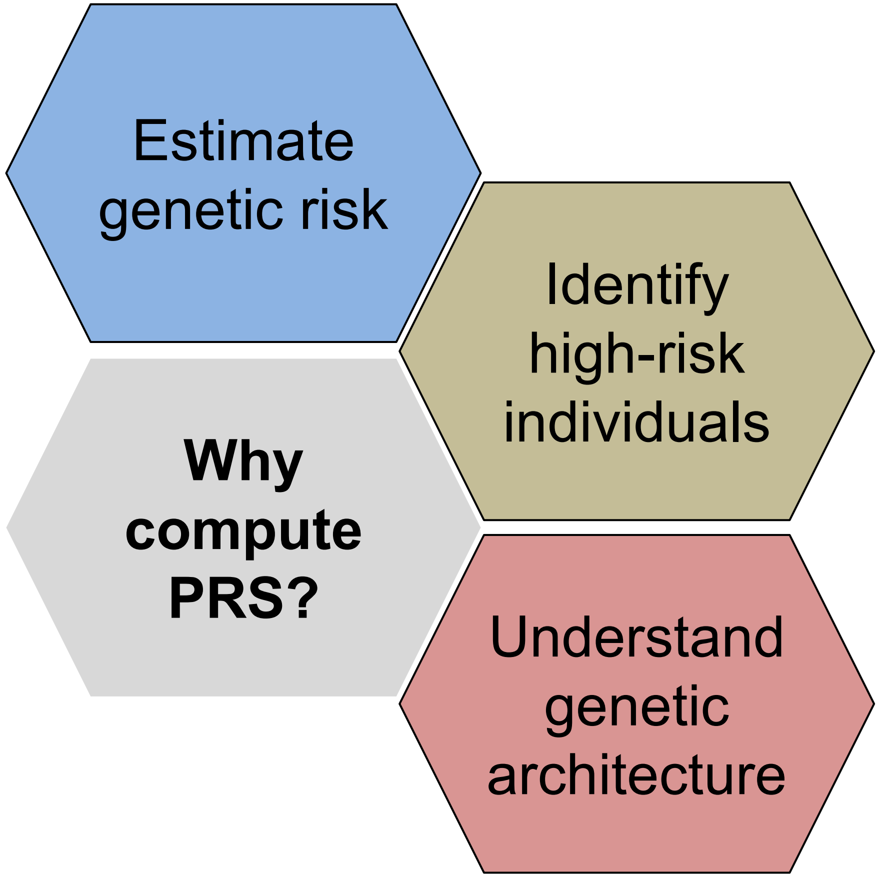
Estimate genetic risk

Identify high-risk individuals

Why compute PRS?

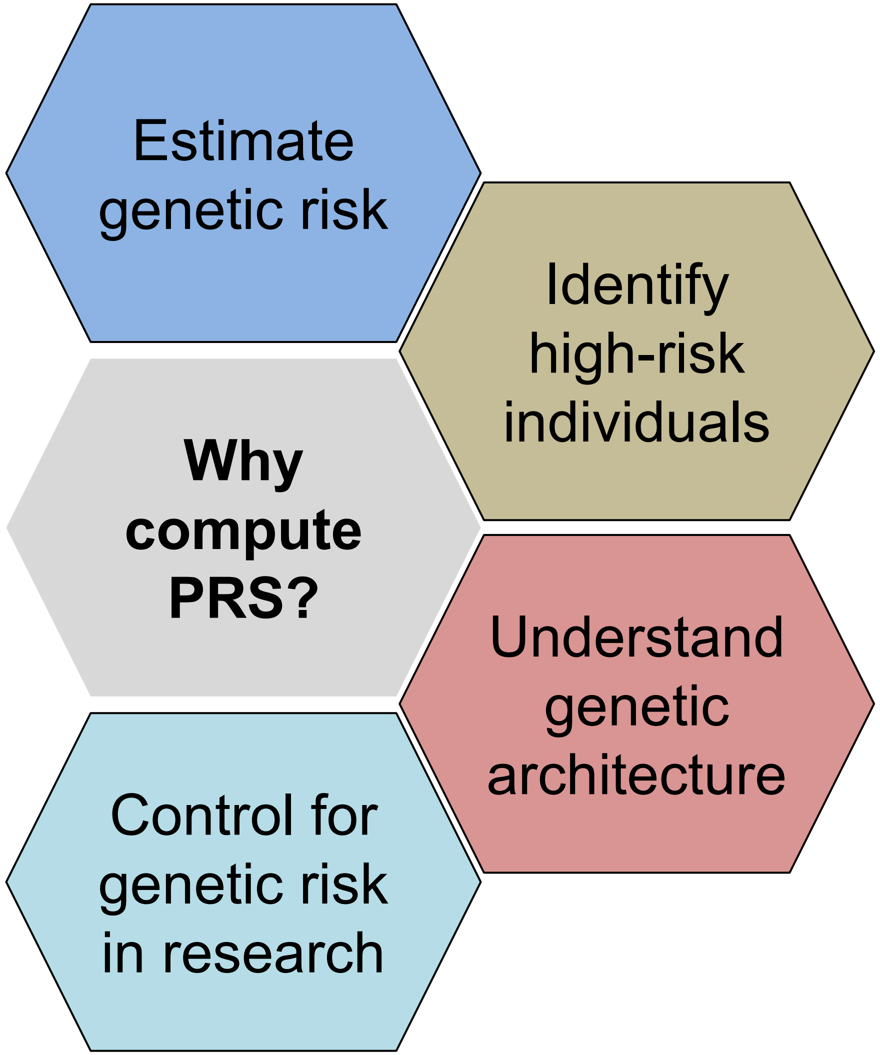
- People with high PRS may have substantially higher risk compared to the population average
- This can help with:
  - ○ Early screening
  - ○ Preventive interventions
  - ○ Personalised medicine

Why use PRS?


Estimate genetic risk
Identify high-risk individuals
Why compute PRS?
Understand genetic architecture

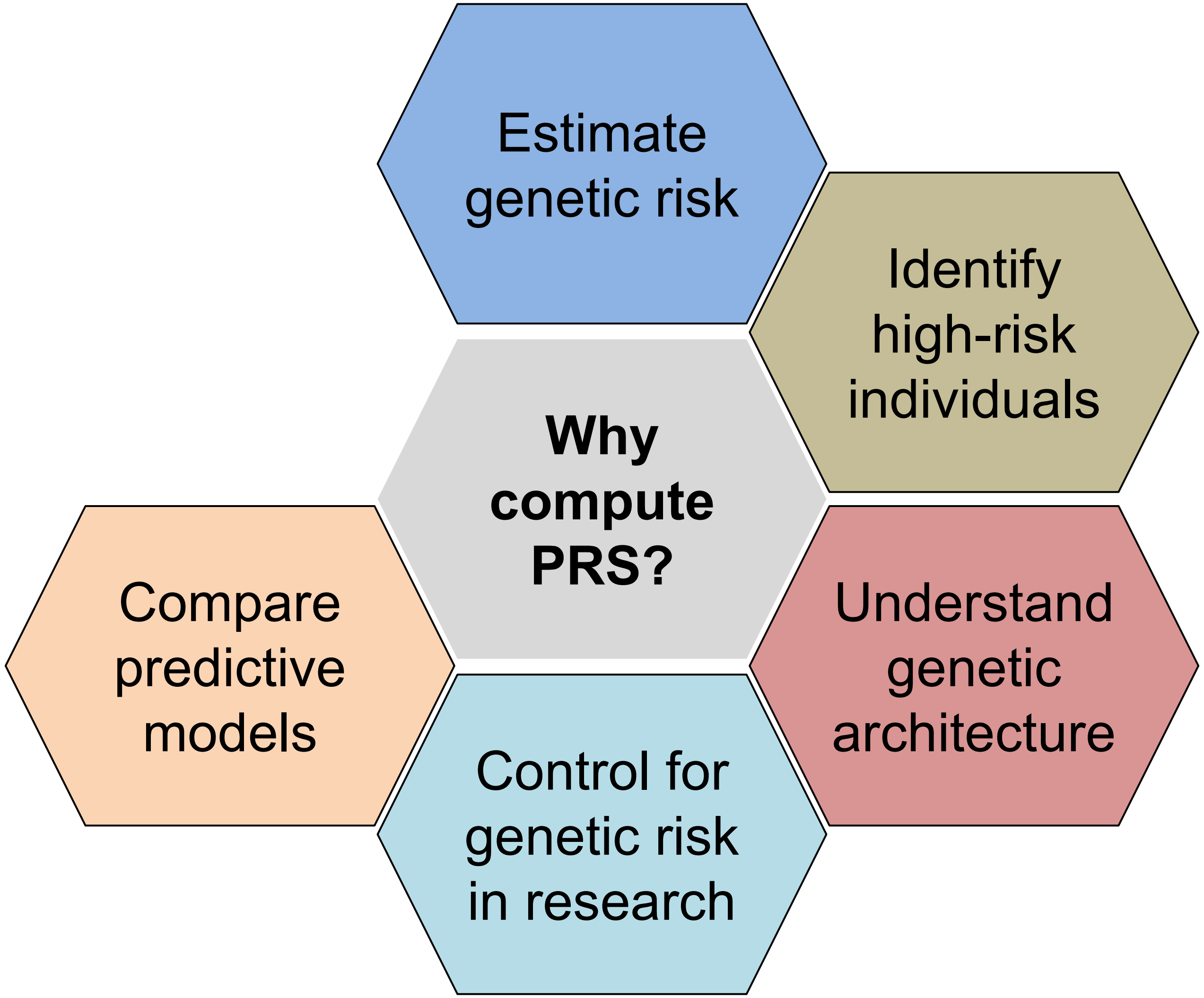
- PRS can help researchers understand how much of a trait is explained by common genetic variation and how different sets of variants contribute to it

Why use PRS?


Estimate genetic risk
Identify high-risk individuals
Why compute PRS?
Understand genetic architecture
Control for genetic risk in research

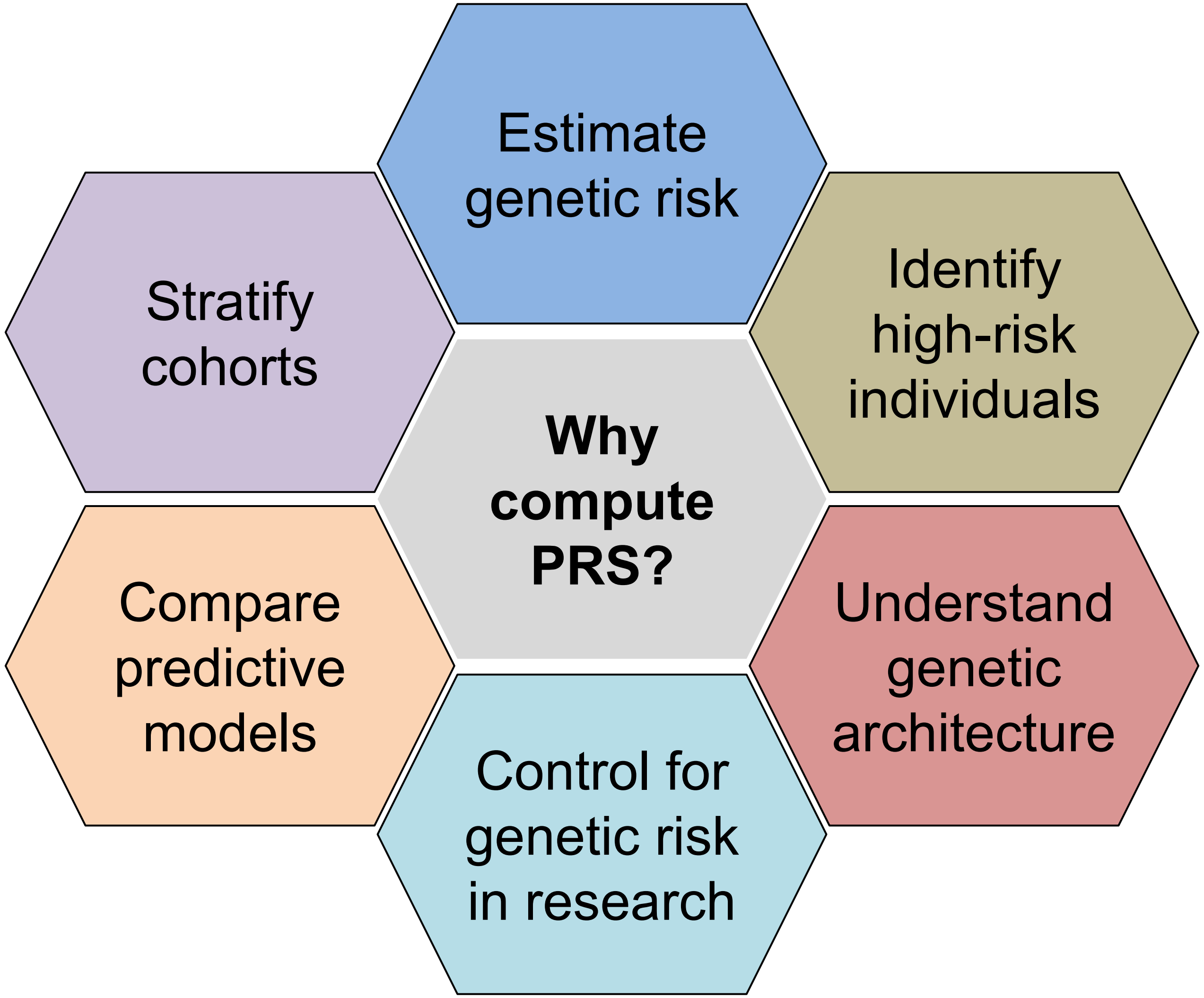
- In studies of e.g. brain imaging, cognition, or biomarkers, PRS can be used as a covariate to control for genetic risk or to explore how genetics influence these traits

Why use PRS?


Estimate genetic risk
Identify high-risk individuals
Why compute PRS?
Compare predictive models
Understand genetic architecture
Control for genetic risk in research

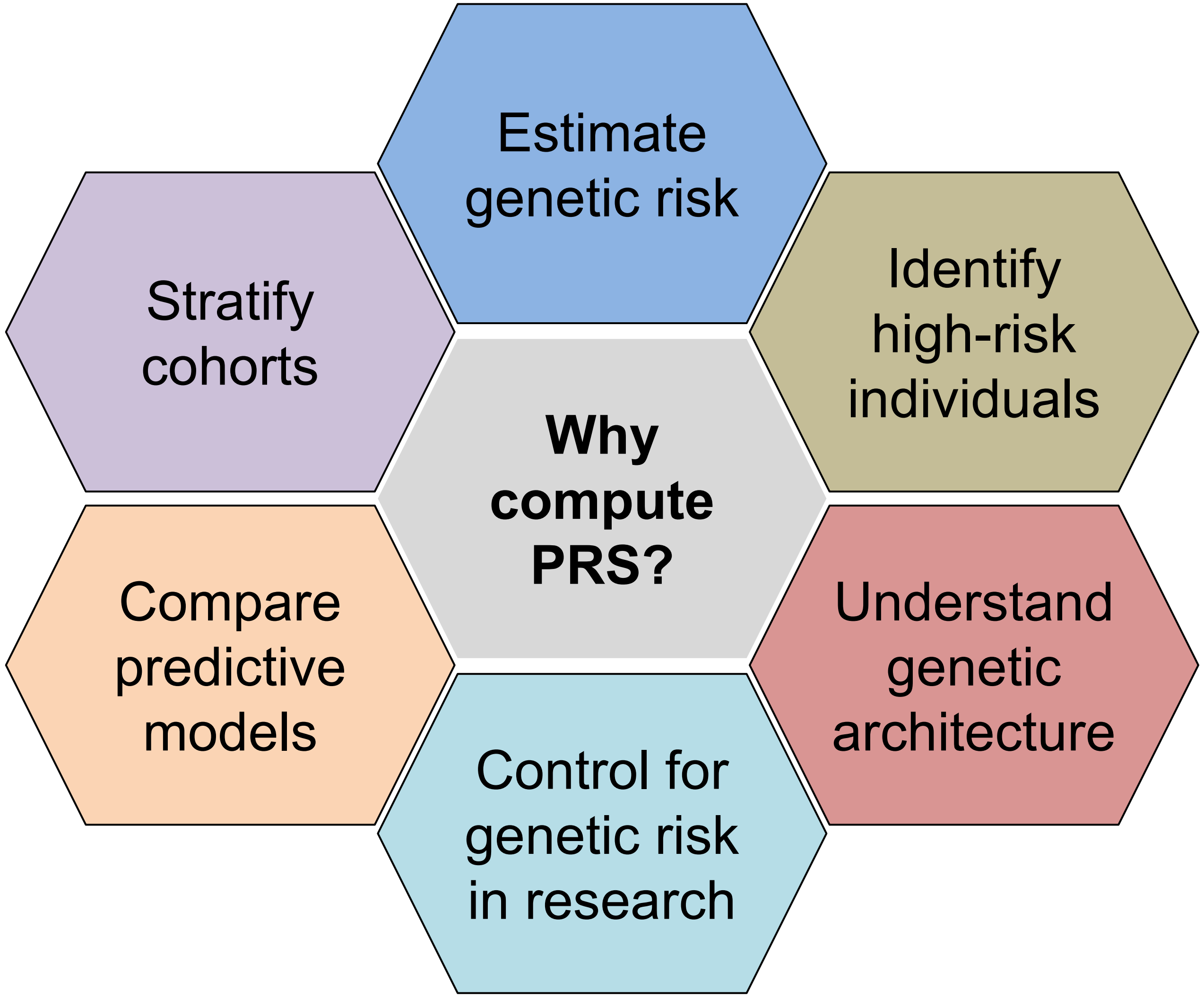
- PRS allows comparison of different genetic models (e.g., including or excluding APOE) to test how well genetic data predict traits

Why use PRS?


Estimate genetic risk
Stratify cohorts
Identify high-risk individuals
Why compute PRS?
Compare predictive models
Understand genetic architecture
Control for genetic risk in research

- PRS can be used to stratify participants into different risk groups, which is especially useful in:
  - Clinical trials
  - Longitudinal studies
  - Prevention research

Why use PRS?


Estimate genetic risk
Stratify cohorts
Identify high-risk individuals
Why compute PRS?
Compare predictive models
Understand genetic architecture
Control for genetic risk in research

- **Important considerations:**
- Ancestry-matched base and target data
- Include population stratification covariates (genetic PCs)
- Validate in independent datasets
- Use multiple thresholds for SNP inclusion (pT)

PRS computation tools

Method	Models LD?	Shrinkage?	Threshold tuning?	Speed	Language	Best use case
LDpred2	✅	✅	❌ (auto)	Moderate	R	Accurate modelling of LD, multiple PRS models
PLINK	❌	❌	✅	✅✅	C++	Fast, large datasets, simple baseline
PRSice	❌	❌	✅ (auto tested)	✅✅	R + C++	Easy, fast PRS screening across thresholds
PRS-CS	✅	✅	❌	Moderate	Python	No threshold tuning, good polygenic modelling
SBayesR	✅	✅	❌	Slow–Moderate	Python/C++	Advanced modelling, large/complex GWAS

PRS computation tools

Method	Models LD?	case
LDpred2	✓	elling of PRS s
PLINK	✗	atasets, eline
PRSice	✗	PRS cross ds
PRS-CS	✓	tuning, jenic ng
SBayesR	✓	delling, GWAS

- GWAS tests each SNP *individually* for association with a trait (e.g., Alzheimer's risk, a beta coefficient and a p-value)
- **Assumption: *each SNP is tested independently***

- **But this independence assumption is not true in reality**

- So LDpred2 aims to recover true SNP effects using a *Bayesian model* that adjusts GWAS effect sizes for linkage disequilibrium (LD)

1. Start with prior beliefs about SNP effect sizes:
    1. Example: Most SNPs have zero effect, a few have small/medium/large effects.
    2. In LDpred2: prior = mixture of Gaussians
2. Use the observed GWAS summary statistics (betas and p-values) and an an external LD matrix (correlations between SNPs from a reference panel)
3. Update the effect size estimates based on:
    1. How strong the GWAS signal is
    2. How correlated that SNP is with others
    3. How likely it is, under the prior, that the SNP has a real effect

- **This process "shrinks" noisy effect sizes towards their true value**

PRS computation tools

Method	Models LD?	Shrinkage?
LDpred2	✓	✓
PLINK	✗	✗
PRSice	✗	✗
PRS-CS	✓	✓
SBayesR	✓	✓

- GWAS effect sizes are often **inflated**, due to:

1. Sampling noise (especially in small samples)
2. Linkage disequilibrium (LD)
3. Winner's curse (top SNPs look stronger than they are)

- **Shrinkage estimates a more conservative value**
- True effects are retained (e.g., SNP1 stays positive) and false positives are shrunk toward zero (e.g., SNP2 is penalised)

- PRS-CS uses *Bayesian continuous shrinkage* that assumes most SNPs have no or small effect and updates the effect sizes using the GWAS data and the LD structure:

1. Keep true effect sizes (even if small)
2. Shrink noisy or false signals toward zero
3. Do all this without altering p-value thresholds

- **Continuous shrinkage process adaptively shrinks the effect sizes based on how strong the GWAS signal is for a particular SNP, and the LD of that SNP with others**

PRS computation tools

Method	Models LD?	Shrinkage?	Threshold tuning?
LDpred2	☑	☑	✗ (auto)
PLINK	✗	✗	☑
PRSice	✗	✗	☑ (auto tested)
PRS-CS	☑	☑	✗
SBayesR	☑	☑	✗

- *p-value thresholding* implemented in PRSice is the process of selecting SNPs based on their GWAS p-values:

1. Setting up a range of p-value thresholds (e.g., 5e-8 to 1.0)

1. Keeping all SNPs within the different GWAS p ≤ threshold

1. Clumping SNPs in LD (to keep independent SNPs) and calculates a PRS for each individual

1. Testing how well different sets of SNPs predict the phenotype using a regression

1. Recording R^2 and p-value and plotting R^2 vs. p-threshold to find the optimal threshold for SNP inclusion

1. Choosing the best-performing threshold (e.g., $p < 0.01$, $p < 0.05$, $p < 0.5$, etc.)

PRS computation tools

Method	Models LD?	Shrinkage?	Threshold tuning?	Speed	Language	Best use case
LDpred2	✅	✅	❌ (auto)	Moderate	R	Accurate modelling of LD, multiple PRS models
PLINK	❌	❌	✅	✅✅	C++	Fast, large datasets, simple baseline
PRSice	❌	❌	✅ (auto tested)	✅✅	R + C++	Easy, fast PRS screening across thresholds
PRS-CS	✅	✅	❌	Moderate	Python	No threshold tuning, good polygenic modelling
SBayesR	✅	✅	❌	Slow–Moderate	Python/C++	Advanced modelling, large/complex GWAS

Input data for PRS calculations with PRSice

GWAS summary statistics = base file
The file with GWAS summary statistics

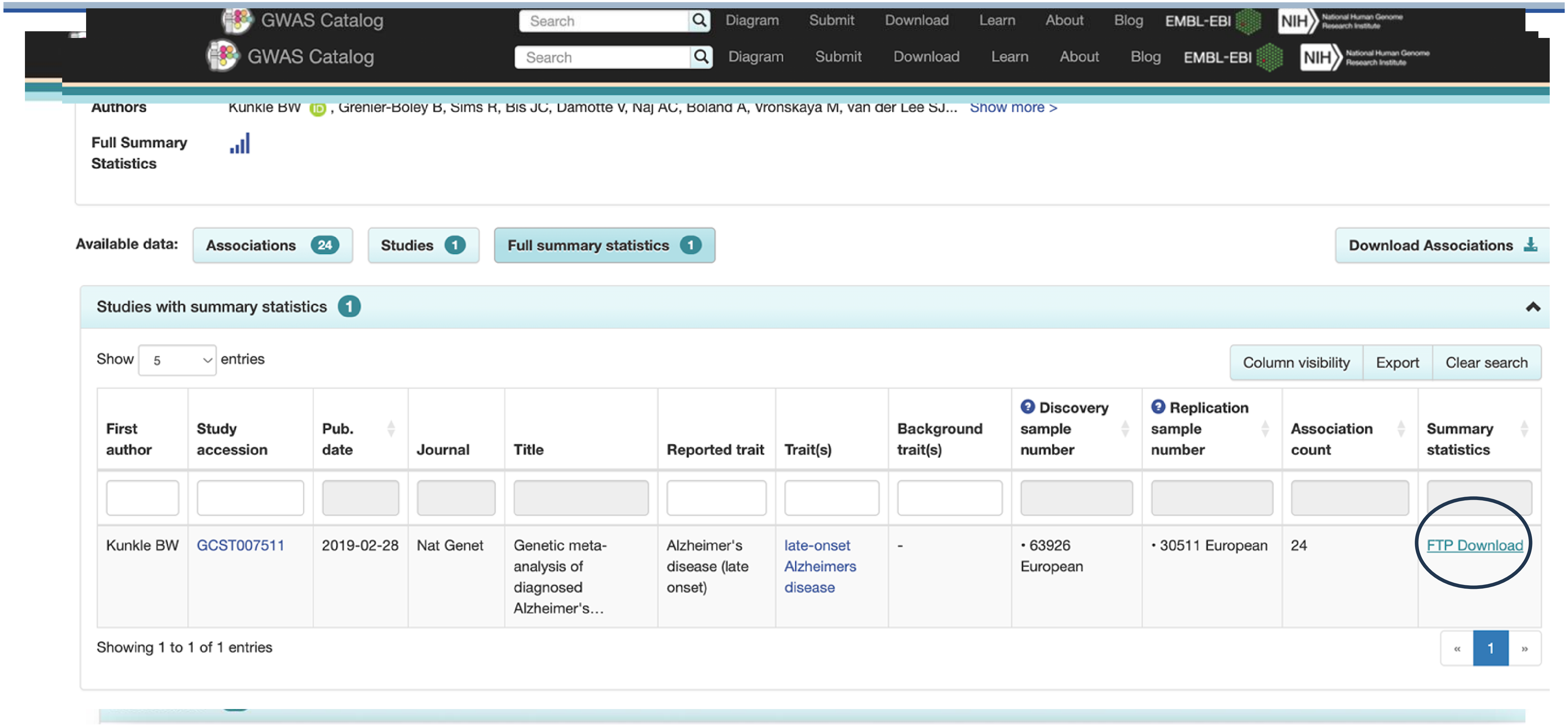
Genotype data = target data
The prefix of the files that contain the genotype data in binary plink format

Optional: Phenotype file
FID – Family ID (usually same as IID if not using family data)
IID – Individual ID
Phenotype – Your target trait or disease status (binary or continuous)

Optional: Covariates file
File containing genetic principal components or other covariates such as age, as necessary

Optional: External dataset for clumping
Within each block of correlated SNPs, the SNP with the lowest p-value in the discovery set is selected

How to obtain GWAS summary statistics: GWAS catalogue example



Authors Kunkle BW, Grenier-Boley B, Sims R, Bis JC, Damotte V, Naj AC, Boland A, Vronskaya M, van der Lee SJ... Show more >

Full Summary Statistics

Available data: Associations 24 | Studies 1 | Full summary statistics 1 | Download Associations

Studies with summary statistics 1

Show 5 entries

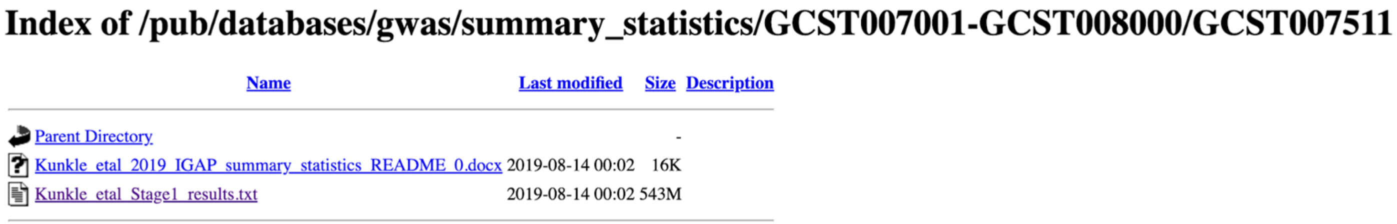
Column visibility | Export | Clear search

First author	Study accession	Pub. date	Journal	Title	Reported trait	Trait(s)	Background trait(s)	Discovery sample number	Replication sample number	Association count	Summary statistics
Kunkle BW	GCST007511	2019-02-28	Nat Genet	Genetic meta-analysis of diagnosed Alzheimer's…	Alzheimer's disease (late onset)	late-onset Alzheimers disease	-	• 63926 European	• 30511 European	24	FTP Download

Showing 1 to 1 of 1 entries

« 1 »

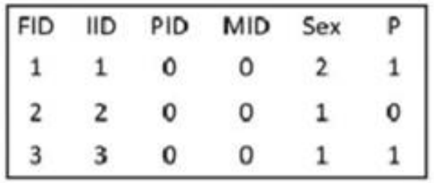
How to obtain GWAS summary statistics: GWAS catalogue example



Index of /pub/databases/gwas/summary_statistics/GCST007001-GCST008000/GCST007511

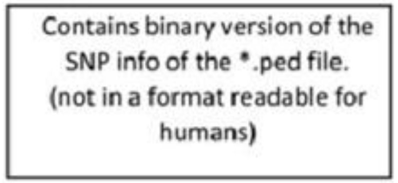
Name	Last modified	Size	Description
Parent Directory		-	
Kunkle_etal_2019_IGAP_summary_statistics_README_0.docx	2019-08-14 00:02	16K	
Kunkle_etal_Stage1_results.txt	2019-08-14 00:02	543M	

*.fam



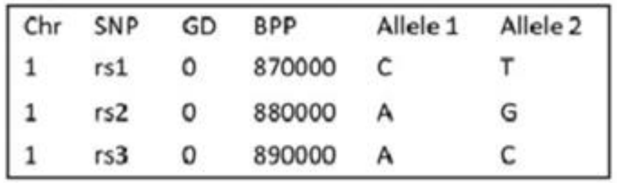
FID	IID	PID	MID	Sex	P
1	1	0	0	2	1
2	2	0	0	1	0
3	3	0	0	1	1

*.bed

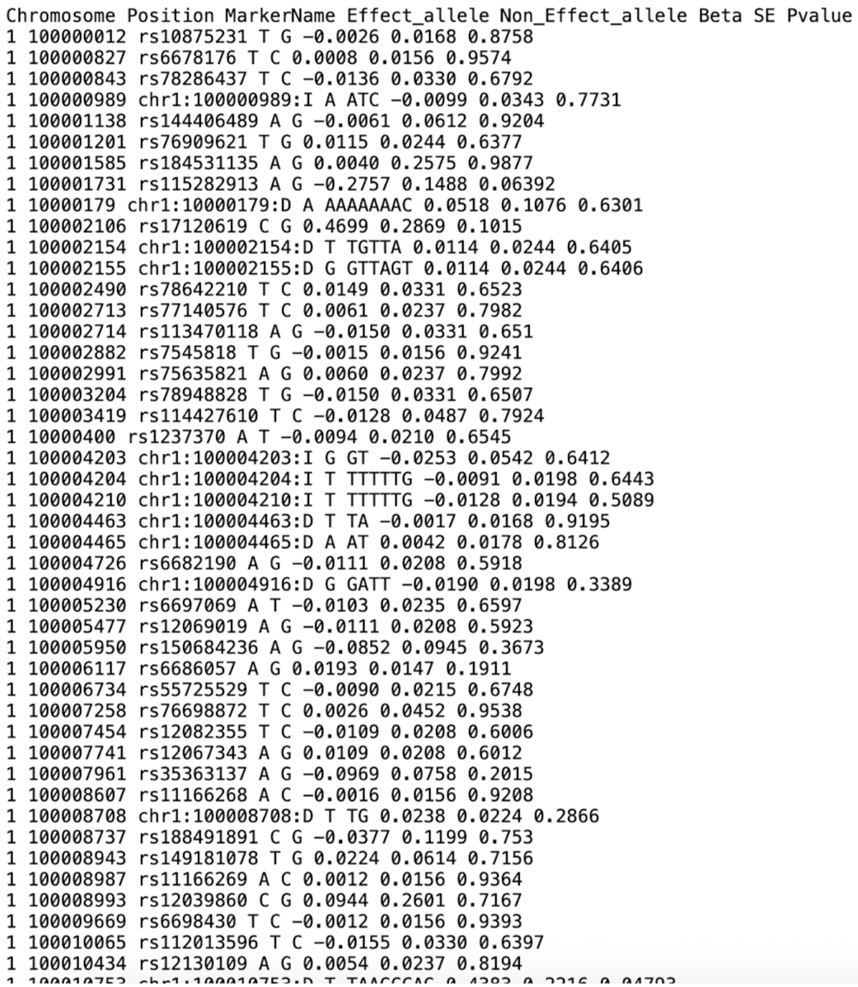


Contains binary version of the SNP info of the *.ped file. (not in a format readable for humans)

*.bim

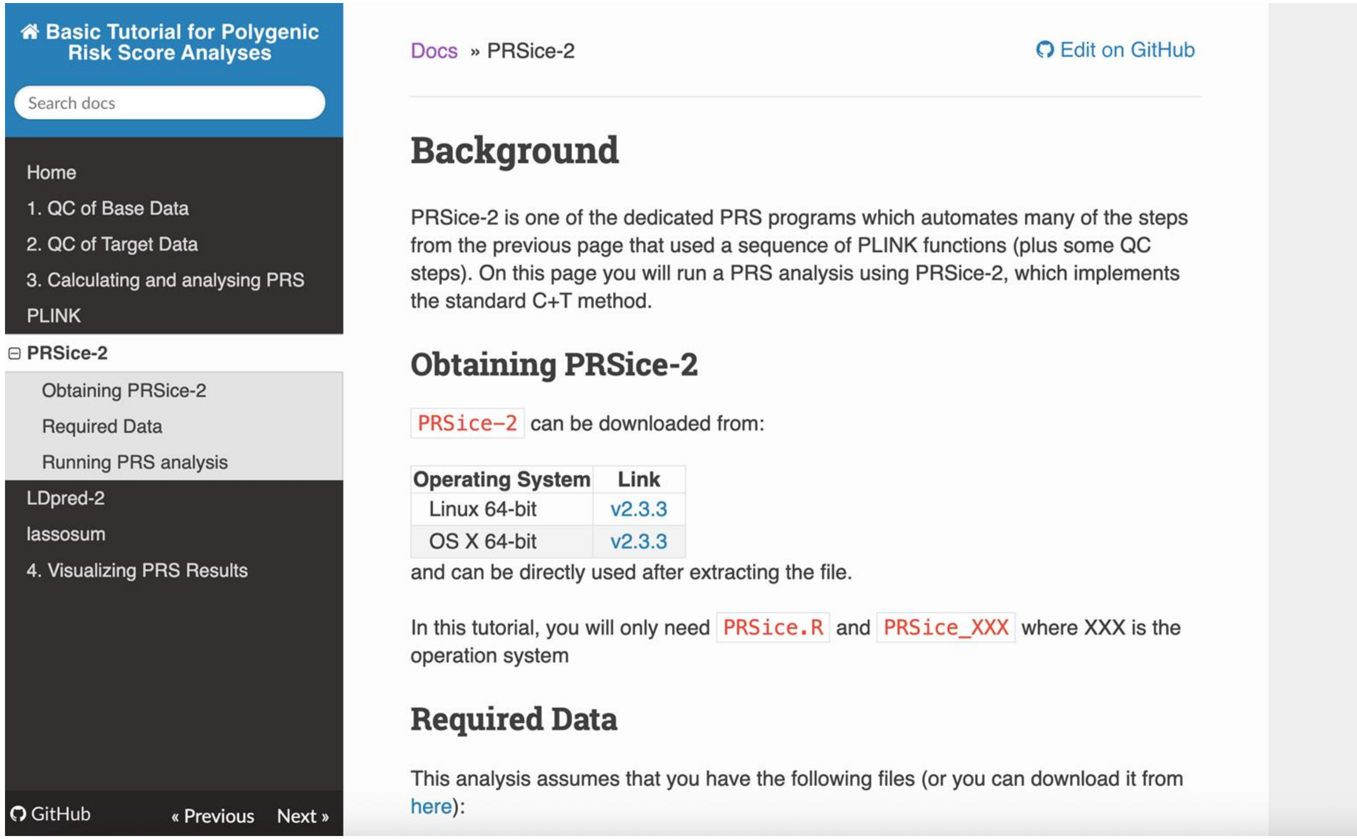


Chr	SNP	GD	BPP	Allele 1	Allele 2
1	rs1	0	870000	C	T
1	rs2	0	880000	A	G
1	rs3	0	890000	A	C



```
Chromosome Position MarkerName Effect_allele Non_Effect_allele Beta SE Pvalue
1 100000012 rs10875231 T G -0.0026 0.0168 0.8758
1 100000827 rs6678176 T C 0.0008 0.0156 0.9574
1 100000843 rs78286437 T C -0.0136 0.0330 0.6792
1 100000989 chr1:100000989:I A ATC -0.0099 0.0343 0.7731
1 100001138 rs144406489 A G -0.0061 0.0612 0.9204
1 100001201 rs76909621 T G 0.0115 0.0244 0.6377
1 100001585 rs184531135 A G 0.0040 0.2575 0.9877
1 100001731 rs115282913 A G -0.2757 0.1488 0.06392
1 10000179 chr1:10000179:D A AAAAAAAC 0.0518 0.1076 0.6301
1 100002106 rs17120619 C G 0.4699 0.2869 0.1015
1 100002154 chr1:100002154:D T TGTTA 0.0114 0.0244 0.6405
1 100002155 chr1:100002155:D G GTTAGT 0.0114 0.0244 0.6406
1 100002490 rs78642210 T C 0.0149 0.0331 0.6523
1 100002713 rs77140576 T C 0.0061 0.0237 0.7982
1 100002714 rs113470118 A G -0.0150 0.0331 0.651
1 100002882 rs7545818 T G -0.0015 0.0156 0.9241
1 100002991 rs75635821 A G 0.0060 0.0237 0.7992
1 100003204 rs78948828 T G -0.0150 0.0331 0.6507
1 100003419 rs114427610 T C -0.0128 0.0487 0.7924
1 10000400 rs1237370 A T -0.0094 0.0210 0.6545
1 100004203 chr1:100004203:I G GT -0.0253 0.0542 0.6412
1 100004204 chr1:100004204:I T TTTTTG -0.0091 0.0198 0.6443
1 100004210 chr1:100004210:I T TTTTTG -0.0128 0.0194 0.5089
1 100004463 chr1:100004463:D T TA -0.0017 0.0168 0.9195
1 100004465 chr1:100004465:D A AT 0.0042 0.0178 0.8126
1 100004726 rs6682190 A G -0.0111 0.0208 0.5918
1 100004916 chr1:100004916:D G GATT -0.0190 0.0198 0.3389
1 100005230 rs6697069 A T -0.0103 0.0235 0.6597
1 100005477 rs12069019 A G -0.0111 0.0208 0.5923
1 100005950 rs150684236 A G -0.0852 0.0945 0.3673
1 100006117 rs6686057 A G 0.0193 0.0147 0.1911
1 100006734 rs55725529 T C -0.0090 0.0215 0.6748
1 100007258 rs76698872 T C 0.0026 0.0452 0.9538
1 100007454 rs12082355 T C -0.0109 0.0208 0.6006
1 100007741 rs12067343 A G 0.0109 0.0208 0.6012
1 100007961 rs35363137 A G -0.0969 0.0758 0.2015
1 100008607 rs11166268 A C -0.0016 0.0156 0.9208
1 100008708 chr1:100008708:D T TG 0.0238 0.0224 0.2866
1 100008737 rs188491891 C G -0.0377 0.1199 0.753
1 100008943 rs149181078 T G 0.0224 0.0614 0.7156
1 100008987 rs11166269 A C 0.0012 0.0156 0.9364
1 100008993 rs12039860 C G 0.0944 0.2601 0.7167
1 100009669 rs6698430 T C -0.0012 0.0156 0.9393
1 100010065 rs112013596 T C -0.0155 0.0330 0.6397
1 100010434 rs12130109 A G 0.0054 0.0237 0.8194
1 100010753 chr1:100010753:D T TAACCCAC 0.4383 0.2216 0.04793
```

Installing PRSice-2



Background

PRSice-2 is one of the dedicated PRS programs which automates many of the steps from the previous page that used a sequence of PLINK functions (plus some QC steps). On this page you will run a PRS analysis using PRSice-2, which implements the standard C+T method.

Obtaining PRSice-2

`PRSice-2` can be downloaded from:

Operating System	Link
Linux 64-bit	v2.3.3
OS X 64-bit	v2.3.3

and can be directly used after extracting the file.

In this tutorial, you will only need `PRSice.R` and `PRSice_XXX` where XXX is the operation system

Required Data

This analysis assumes that you have the following files (or you can download it from here):

```
wget https://github.com/choishingwan/PRSice/releases/download/2.2.11/PRSice_linux.nightly.zip
unzip PRSice_linux.nightly.zip

Rscript PRSice.R --dir .
```

Virtual machine

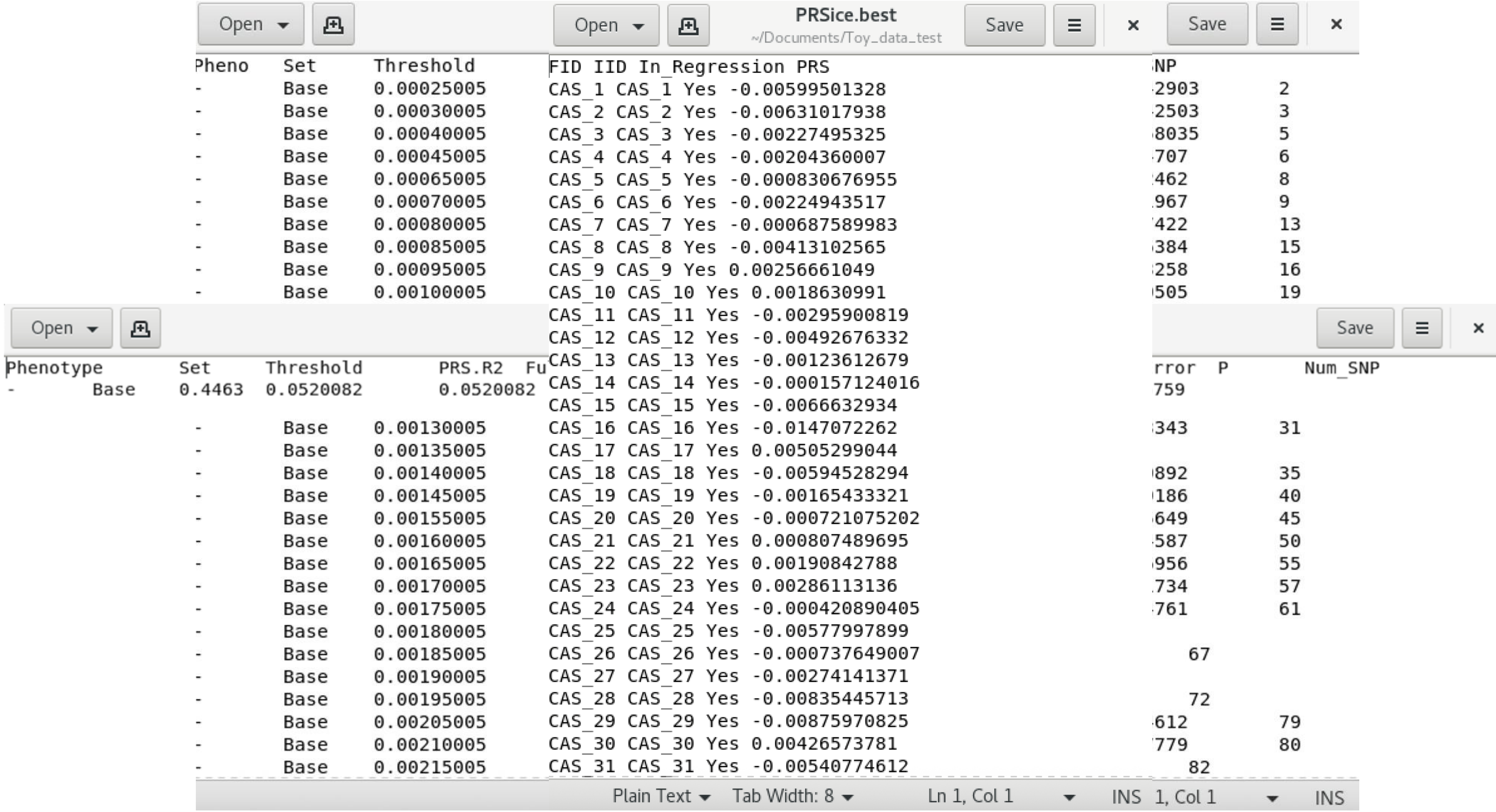
- cd to home directory in terminal
- Navigate to PRSice directory to note the locations of the base and target datasets
- Use the following code to run PRS computation, ensure that you copy the correct locations of the files

Running PRSice-2: Example Command

```
Rscript /home/as2-streaming-user/PRSice/PRSice.R --dir . \
--prsice /home/as2-streaming-user/PRSice/PRSice_linux \
--base /home/as2-streaming-user/PRSice/TOY_BASE_GWAS.assoc  \
--target /home/as2-streaming-user/PRSice/TOY_TARGET_DATA \
--thread 1  \
--stat OR \
--binary-target T
```

PRSice output Files

1. **.log**
   ○ Log file with all information regarding the computation
2. **.prsice**
   ○ Information about the number of SNPs in each score computed
3. **.summary**
   ○ Provides information about the best-fit PRS
4. **.best**
   ○ Contains PRS for each individual at the best-fit PRS name



```
Pheno   Set     Threshold
-       Base    0.00025005
-       Base    0.00030005
-       Base    0.00040005
-       Base    0.00045005
-       Base    0.00065005
-       Base    0.00070005
-       Base    0.00080005
-       Base    0.00085005
-       Base    0.00095005
-       Base    0.00100005

-       Base    0.00130005
-       Base    0.00135005
-       Base    0.00140005
-       Base    0.00145005
-       Base    0.00155005
-       Base    0.00160005
-       Base    0.00165005
-       Base    0.00170005
-       Base    0.00175005
-       Base    0.00180005
-       Base    0.00185005
-       Base    0.00190005
-       Base    0.00195005
-       Base    0.00205005
-       Base    0.00210005
-       Base    0.00215005
```

```
Phenotype       Set     Threshold       PRS.R2    Fu
-       Base    0.4463  0.0520082       0.0520082
```

```
rror  P        Num_SNP
759
```

```
FID IID In_Regression PRS
CAS_1 CAS_1 Yes -0.00599501328
CAS_2 CAS_2 Yes -0.00631017938
CAS_3 CAS_3 Yes -0.00227495325
CAS_4 CAS_4 Yes -0.00204360007
CAS_5 CAS_5 Yes -0.000830676955
CAS_6 CAS_6 Yes -0.00224943517
CAS_7 CAS_7 Yes -0.000687589983
CAS_8 CAS_8 Yes -0.00413102565
CAS_9 CAS_9 Yes 0.00256661049
CAS_10 CAS_10 Yes 0.0018630991
CAS_11 CAS_11 Yes -0.00295900819
CAS_12 CAS_12 Yes -0.00492676332
CAS_13 CAS_13 Yes -0.00123612679
CAS_14 CAS_14 Yes -0.000157124016
CAS_15 CAS_15 Yes -0.0066632934
CAS_16 CAS_16 Yes -0.0147072262
CAS_17 CAS_17 Yes 0.00505299044
CAS_18 CAS_18 Yes -0.00594528294
CAS_19 CAS_19 Yes -0.00165433321
CAS_20 CAS_20 Yes -0.000721075202
CAS_21 CAS_21 Yes 0.000807489695
CAS_22 CAS_22 Yes 0.00190842788
CAS_23 CAS_23 Yes 0.00286113136
CAS_24 CAS_24 Yes -0.000420890405
CAS_25 CAS_25 Yes -0.00577997899
CAS_26 CAS_26 Yes -0.000737649007
CAS_27 CAS_27 Yes -0.00274141371
CAS_28 CAS_28 Yes -0.00835445713
CAS_29 CAS_29 Yes -0.00875970825
CAS_30 CAS_30 Yes 0.00426573781
CAS_31 CAS_31 Yes -0.00540774612
```

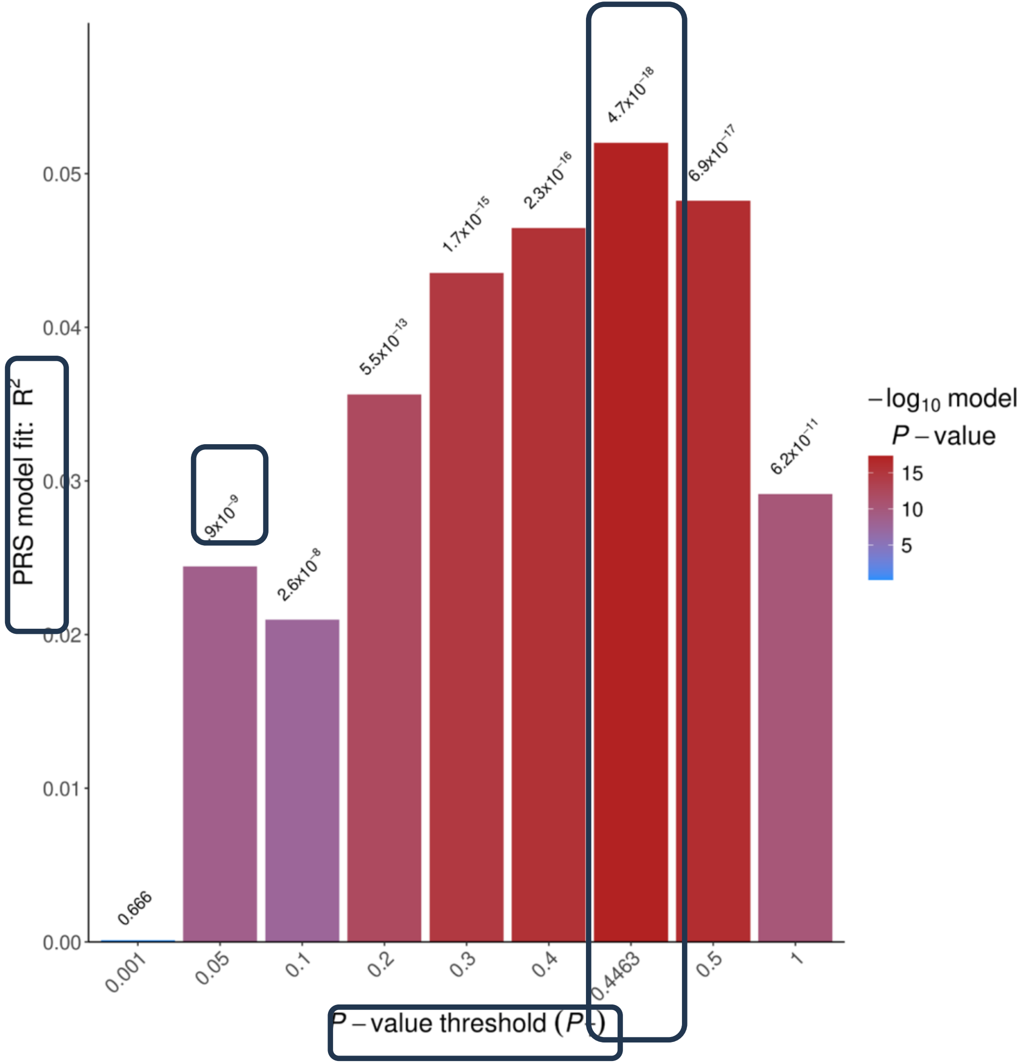
```
NP
2903        2
2503        3
8035        5
707         6
462         8
967         9
422         13
384         15
258         16
505         19

343         31

892         35
186         40
649         45
587         50
956         55
734         57
761         61

   67

   72
612         79
779         80
   82
```

PRSice output files

- The first plot is PRSice_BARPLOT_<date>.png

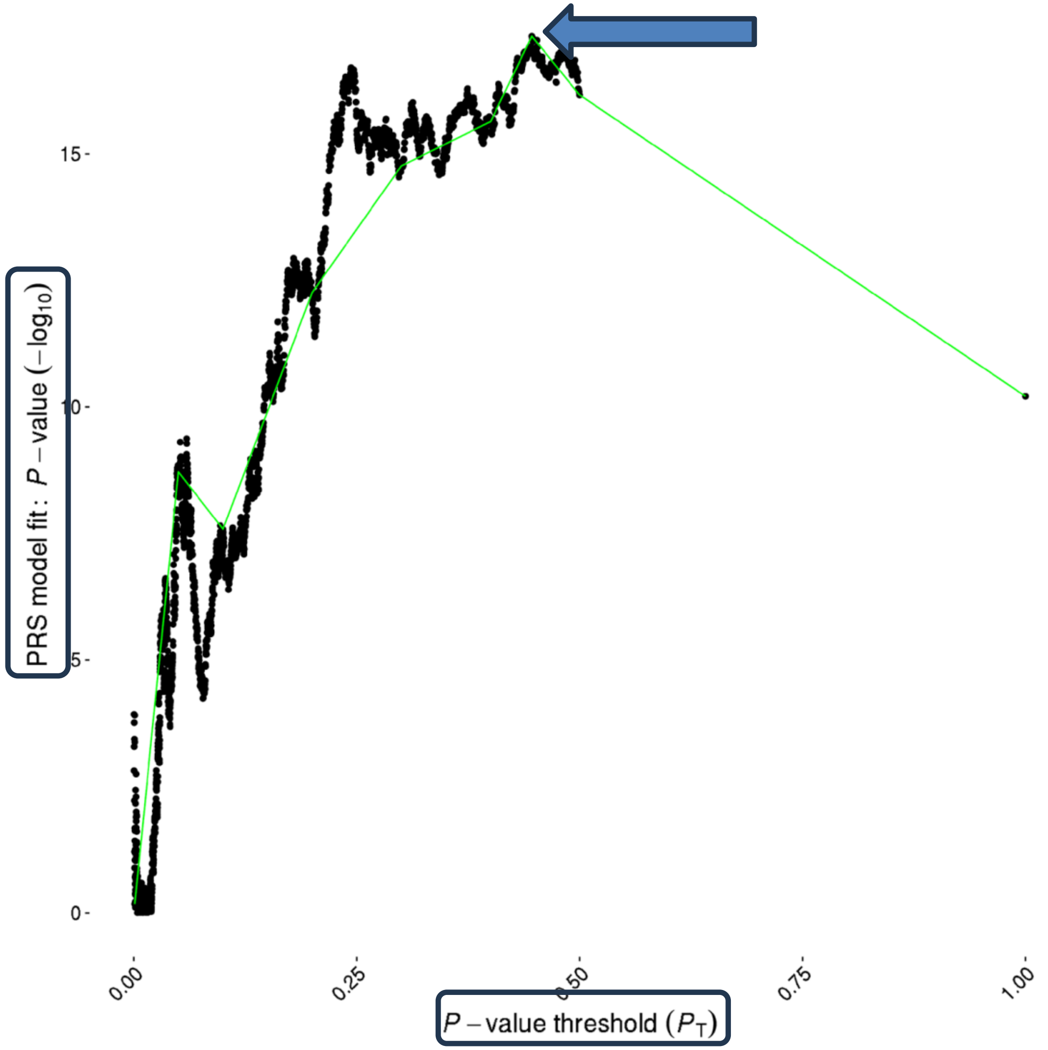
- X-axis = p-value threshold for SNP inclusion (pT)
- Y-axis = predictive value, Nagelkerke’s R^2
- Each bar shows the model p-value

- Using SNPs with a p-value up to 0.4463 achieves the highest predictive value in the target sample with a p-value of 4.7e-18



PRSice output files

- The second plot is PRSice_HIGH-RES_PLOT_<date> .png
- X-axis = p-value threshold for SNP inclusion (pT)
- Y-axis = PRS p-values
- The p-value of the predictive effect is in black together with an aggregated trend line in green
- Of note: PRS analysis typically shows that models with lenient p-value thresholds often predict better than models with more stringent thresholds, suggesting that many statistically insignificant SNPs still have predictive value in polygenic traits



Example: PRS in Alzheimer’s Disease, ADNI

```
Rscript /home/as2-streaming-user/PRSice/PRSice.R --dir . \
--prsice /home/as2-streaming-user/PRSice/PRSice_linux \
--base /home/as2-streaming-user/data/GCST90027158_buildGRCh38.tsv \
--target /home/as2-streaming-user/data/ADNI_QC_FINAL \
--thread 1  \
--snp variant_id \
--chr chromosome \
--bp base_pair_location \
--A1 effect_allele \
--A2 other_allele \
--stat beta \
--pvalue p_value \
--bar-levels 5e-8,1e-5,0.1 \
--binary-target T \
--fastscore T \
--out ADNI_PRS
```

Analysis with global CL burden

- See word document