GPRS

Ying-Chu Lo

Mar 02, 2023

CONTENTS:

CHAPTER

ONE

ABOUT GPRS PACKAGE

This package aims to generate the PRS model from GWAS summary statistics. It is designed to deal with the data format based on the GWAS catalog and GeneATLAS database GWAS summary statistics data.

- Understanding the workflow of this package:

1. Filter GWAS summary statistics files (remove duplicate SNPID and select significant SNPs by P-value)
2. Generate bfiles by Plink1.9
3. Do clumping by Plink1.9
4. Generate PRS model by Plink2.0
5. Calculate statistic value of PRS model

1.1 Environment setup

1. Setup virtualenv

```
$ python3 -m venv venv
```

2. Activate virtualenv

```
$ source ./venv/bin/activate
```

3. Install this package

```
$ pip install -e .
```

- Twelve commands in gprs:

1. `geneatlas-filter-data`
2. `gwas-filter-data`
3. `generate-plink-bfiles`
4. `clump`
5. `select-clump-snps`
6. `build-prs`
7. `combine-prs`
8. `prs-statistics`

9. `combine-prs-stat`
10. `transfer_atcg` (optional)
11. `sub-setpop` (optional)
12. `generate_plink_bfiles_w_individual_info` (optional)
13. `random_draw_samples_from_fam` (optional)
14. `subset_vcf_w_random_sample` (optional)

CHAPTER

TWO

GET STARTED: UNDERSTANDING THE DATA FORMAT

2.1 GWAS template

2.2 GeneAtlas template

2.3 How to choose model template?

CHAPTER

THREE

BEFORE STEP1:

Please unzip your .gz file first.

CHAPTER

FOUR

STEP1: UNIFY THE DATA FORMAT

After knowing the data format, users can choose the model (gwas or geneatlas) to unify the data format and filter out SNPs(optional). :heavy_exclamation_mark: SNPs are extract out by RSID not chromosome position

4.1 Function: `gprs geneatlas-filter-data`

Filter GeneAtlas csv file by P-value and unify the data format as following order: SNPID, ALLELE, BETA, StdErr, Pvalue

4.2 How to use it?

Shell:

```
$ gprs geneatlas-filter-data --ref [str] --data_dir [str] --result_dir [str] --snp_id_
↪header [str] --allele_header [str] --beta_header [str] --se_header [str] --pvalue_
↪header [str] --pvalue [float/scientific notation] --output_name [str]
$ gprs gwas-filter-data --ref [str] --data_dir [str] --result_dir [str] --snp_id_header␣
↪[str] --allele_header  [str] --beta_header [str] --se_header [str] --pvalue_header␣
↪[str] --pvalue [float/scientific notation] --output_name [str]
```

Python:

```
from gprs.gene_atlas_model import GeneAtlasModel
if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.filter_data( snp_id_header='MarkerName',
                           allele_header='Allele1',
                           beta_header='b',
                           se_header ='SE',
                           pvalue_header='p',
                           output_name='2014height')

from gprs.gwas_model import GwasModel
if __name__ == '__main__':
    gwas = GwasModel( ref='/home1/ylo40816/1000genomes/hg19',
                  data_dir='/home1/ylo40816/Projects/GPRS/data/2019_GCST008970')
```

(continues on next page)

(continued from previous page)

```
gwas.filter_data( snp_id_header='RSID',
                  allele_header='Allele1',
                  beta_header='Effect',
                  se_header='StdErr',
                  pvalue_header='P-value',
                  output_name='GCST008970',
                  file_name='gout_chr1_22_LQ_IQ06_mac10_all_201_rsid.csv')
```

4.3 output files

- `*.QC.csv` (QC files)
- `*.csv` (snplist)

CHAPTER

FIVE

SETP 2: GENERATE PLINK BFILES (.BIM, .BAM, .FAM)

After filtering SNPs in GWAS summary statistics data. Users have to generate plink files for generate PRS model.

5.1 Function: `gprs generate-plink-bfiles`

This option encodes plink1.9 make-bed function

```
plink --vcf [ref] --extract [snplists after qc] --make-bed --out [bfile folder/output_
↪name]
```

5.2 How to use it?

Shell:

```
$ gprs generate-plink-bfiles --ref [str] --snplist_name [str] --symbol [str] --output_
↪name [str]
```

Python:

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.generate_plink_bfiles(snplist_name='2014height_MEC', output_name=
↪'2014height_hg38',extra_commands="--vcf-half-call r" ,symbol='_GRCh38.genotypes')
```

5.3 output files

- `*.bim`
- `*.bed`
- `*.fam`

CHAPTER

SIX

SETP 3-1: CLUMPING (REMOVE LINKED SNPS)

6.1 Function: `gprs clump`

This option encodes plink1.9 clump function

```
plink --bfile [bfiles] --clump [qc snpslists] --clump-p1  --clump-p2  --clump-r2  --
↪clump-kb  --clump-field  --clump-snp-field  --out
```

The plink_bfiles_dir, qc snpslists and clump_output_dir will automatically be filled in the script. Users have to indicate the options below.

6.2 How to use it?

Shell:

```
$ gprs clump --ref [str] --data_dir [str] --clump_kb [int] --clump_p1 [float/scientific␣
↪notation] --clump_p2 [float/scientific notation] --clump_r2 [float] --clump_field␣
↪[str] --clump_snp_field [str] --plink_bfile_name [str] --qc_file_name [str] --output_
↪name [output name]
```

Python:

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.clump(output_name='2014height',
                    clump_kb='250',
                    clump_p1='0.02', clump_p2='0.02',
                    qc_file_name='2014height',
                    plink_bfile_name='2014height')
```

6.3 output files

- `*.clump`

CHAPTER

SEVEN

SETP 3-2: FILTER SNPS DEPENDS ON .CLUMP

After clumping, we have to filter SNPs again, to remove linked SNPs. In this step, we will have new SNPs list, and use it for generate PRS model.

7.1 Function: gprs select-clump-snps

7.2 How to use it?

Shell:

```
$ gprs select-clump-snps --result_dir [str] --qc_file_name [str] --clumpfolder_name␣
↪[str] --clump_file_name [str] --clump_kb [int] --clump_p1 [float/scientific notation] -
↪-clump_r2 [float] --output_name [output name]
```

Python:

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.select_clump_snps(output_name='2014height',clump_file_name='2014height',
                          qc_file_name='2014height',clumpfolder_name='',clump_kb='250',
                    clump_p1='0.02', clump_r2='0.02')
```

7.3 output files

- *.qc_clump_snpslist.csv

7.4 With Chromosome information:

7.5 Without Chromosome information:

CHAPTER

EIGHT

SETP 4-1: GENERATE PRS MODEL

Generate PRS model by using Dosage by plink2.0

8.1 Function: gprs build-prs

```
plink2 --vcf [vcf input] dosage=DS --score [snplists afte clumped and qc]  --out
```

The clumped qc snpslists and prs_output_dir will automatically be filled in the script. Users have to indicate the options below.

8.2 How to use it?

Shell:

```
$ gprs build-prs --vcf_input [str] --qc_clump_snplist_foldername [str] --symbol [str/
↪int] --columns [int] --plink_modifier [str] --memory [int] --clump_kb [int] --clump_p1␣
↪[float/scientific notation] --clump_r2 [float] --output_name [output name]
```

Python:

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.build_prs( vcf_input= '1000genomes/hg19',
                         output_name ='2014height', memory='1000',clump_kb='250',
                    clump_p1='0.02', clump_r2='0.02', qc_clump_snplist_foldername=
↪'2014height')
```

8.3 output files

- `*.sscore`

CHAPTER

NINE

SETP 4-2: COMBINED PRS MODEL

Combined PRS model (python script create by Soyoung Jeon; update by Ying-Chu Lo))

9.1 Function:`gprs combine-prs`

Combine-prs will combine all .sscore files as one .sscore file. And calculate score average and sum per individual.

9.2 How to use it?

Shell:

```
$ gprs combine-prs --ref [str] --result_dur [str]
```

Python:

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.combine_prs(filename="2014height",clump_r2="0.5",clump_kb="250",clump_p1=
↪"0.02")
```

9.3 output files

- `*.sscore`

CHAPTER

TEN

SETP 5-1: CALCULATE STATISTICS VALUE OF PRS MODEL

(Rscript create by Soyoung Jeon; update by Ying-Chu Lo)

10.1 Function: `prs-statistics`

After obtained combined sscore file, `prs-statistics` calculates BETA, AIC, AUC, PseudoR2 and OR ratio

10.2 How to use it?

Shell:

```
$ gprs prs-statistics --score_file [str] --pheno_file [str] --data_set_name [str] --prs_
↪stats_R [str] --r_command [str] --output_name [str]
```

Python:

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.prs_statistics(output_name='2014height', score_file = "/home1/ylo40816/
↪Projects/GPRS/tmp/2014height_250_0.02_0.1.sscore",
        pheno_file = "Projects/GPRS/tmp/result/plink/prs/2014height_pheno.csv",
        r_command='/spack/apps/linux-centos7-x86_64/gcc-8.3.0/r-4.0.0-
↪jfy3icn4kexk7kyabcoxuio2iyyww3o7/bin/Rscript',
        prs_stats_R="Projects/GPRS/gprs/prs_stats_quantitative_phenotype.R", data_set_
↪name="2014height",
                            clump_kb='250',
                            clump_p1='0.02',
                            clump_r2='0.1'
                            )
```

10.3 output files

- `*_stat.txt`

CHAPTER

ELEVEN

SETP 5-2: COMBINED PRS STATISTICS RESULTS (OPTIONAL)

11.1 Function:`combine-prs-stat`

If you have more than one trained PRS model, `combine-prs-stat` function is designed to combine statistics results. For instance: the first PRS model was filtered with P < 0.05, the second PRS model was filtered with P < 0.0005. You will have DATA_0.05_stat.txt/DATA_0.0005_stat.txt Combining two statistic tables allows users easy to compare between PRS models

11.2 How to use it?

Shell:

```
$ gprs combine-prs-stat --data_set_name [str]
```

Python:

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.combine_prs_stat(data_set_name='2014height')
```

11.3 output files

- `*_combined_stat.txt`

CHAPTER

TWELVE

EXAMPLE OF GIANT HEIGHT FROM WOOD *ET AL* 2014

This is an example from Wood et al. The example aimed to use the GPRS package to replicate the Fig4A in Wood *et al* paper.

12.1 Get started: 2014 height data structure

The GWAS summary statistics were downloaded from the GIANT database.

- Data name: GIANT_HEIGHT_Wood_et_al_2014_publicrelease_HapMapCeuFreq.txt

12.2 The data structure:

12.3 Which template to use?

Before starting to process the data, we have to choose which template to use. Please use the table below to select the template.

- The chromosome information is absent in 2014 height data, and 2014 height data also has no header with Chromosome information. Thus, I choose `gene_atlas_model` as a template

12.4 Output folders

In the GPRS package, the result folder will automatically generate under the execution directory. i.e. The user run GPRS package in `/home/user/` then the default result directory is `/home/user/result`

The structure of output folders are: `result/plink` `result/qc` `result/snplists` `result/stat` `result/plink/bfiles` `result/plink/clump` `result/plink/prs` `result/plink/qc_and_clump_snpslist`

12.5 Step1 preparing data set - unified the data format

In step one, the `filter_data` function will filter raw data with `p-value`, the default is 1 and gives you three output files: `snplist`, `qc'd file` and `summary` After preparing the data, the snplist: `result/snplists/[OUTPUT_NAME].csv` will be used to generate plink files.

The qc'd file `result/qc/[OUTPUT_NAME].QC.csv` is a unified file header to `|SNPID| Allele| Beta| SE |Pvalue|` and this file will be used in the clumping step. The summary file records the information of the SNPs number before and after data preparation.

```
$ gprs geneatlas-filter-data --ref [str] --data_dir [str] --result_dir [str] --snp_id_
↪header [str] --allele_header [str] --beta_header [str] --se_header [str] --pvalue_
↪header [str] --pvalue [float/scientific notation] --output_name [str]
```

- In real use:

```
$ gprs geneatlas-filter-data --data_dir data/2014_GWAS_Height --result_dir [str] --snp_
↪id_header MarkerName --allele_header Allele1 --beta_header b --se_header SE --pvalue_
↪header p --pvalue 1 --output_name 2014height
```

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.filter_data( snp_id_header='MarkerName',
                           allele_header='Allele1',
                           beta_header='b',
                           se_header ='SE',
                           pvalue_header='p',
                           output_name='2014height')
```

12.6 After preparing the data-set three files obtained

The output folder will automatically generate and named as `result`

- `result/snplists/2014height.csv`
- `result/qc/2014height.QC.csv`
- summary file: `/result/qc/[output_name].filteredSNP.withPvalue0.05.summary.txt`

12.7 Step2 use qc'd SNPs list to obtain Plink bfiles(.bed, .bim, .fam)

Step2 uses the SNP list from step1 to generate plink bfiles.

```
$ gprs generate-plink-bfiles --ref [str] --snplist_name [str] --output_name [str] --
↪symbol [str] --extra_commands [str]
```

- In real use:

```
$ gprs generate-plink-bfiles --ref 1000genomes/hg19 --snplist_name 2014height --symbol .␣
↪--output_name 2014height
```

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.generate_plink_bfiles(snplist_name='2014height', output_name='2014height',
↪extra_commands="--vcf-half-call r" ,symbol='.genotypes')
```

- chr1-chr22 bfiles obtained

1. `result/plink/bfiles/chr(1-22)_2014height.bed`
2. `result/plink/bfiles/chr(1-22)_2014height.bim`
3. `result/plink/bfiles/chr(1-22)_2014height.fam`

12.8 Step3.1 remove linked SNPs

In step3, we are trying to find the linked SNPs and remove them for further analysis. The package uses the plink `clump` function to find the linked SNPs. The `r2` value in the `clump` function is `0.1`; if the user wants to apply another value, it should be specified in the command. The function `--clump_field` is asking the user to enter the column name, the default here is `Pvalue` (from the Step1 `qc'd` output)

```
$ gprs clump --plink_bfile_name [str] --output_name [str] --clump_kb [int] --clump_p1␣
↪[float/scientific notation] --clump_p2 [float/scientific notation] --clump_r2 [float] -
↪-clump_field [str] --qc_file_name [str] --clump_snp_field [str]
```

- In real use:

```
$ gprs clump --data_dir data/2014_GWAS_Height --clump_kb 250 --clump_p1 0.02 --clump_p2␣
↪0.02 --clump_r2 0.1 --clump_field Pvalue --clump_snp_field 2014height --plink_bfile_
↪name 2014height --qc_file_name 2014height --output_name 2014height
```

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.clump(output_name='2014height',plink_bfile_name='2014height',
```

(continues on next page)

(continued from previous page)

```
                    qc_file_name='2014height',clump_kb='250',
                    clump_p1='0.02',clump_p2='0.02', clump_r2='0.02')
```

- chr1-chr22 clumped files obtained

`result/plink/clump/*.clumped`

12.9 Step3.2 select clumped SNPs

After clump, we will receive a list of SNPs, and we filter out the original qc'd file to generate a new SNPs list. From this step, users have to provide C+T (clumping + threshold) conditions as a marker in the output file name.

```
$ gprs select-clump-snps --qc_file_name [str] --clump_file_name [str] --output_name␣
↪[output name] --clump_kb [int] --clump_p1 [float/scientific notation] --clump_r2␣
↪[float] --clumpfolder_name [str]
```

- In real use:

```
$ gprs select-clump-snps --qc_file_name 2014height --clump_file_name 2014height --clump_
↪kb 250 --clump_p1 0.02 --clump_r2 0.1 --clumpfolder_name 2014height --output_name␣
↪2014height
```

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.select_clump_snps(output_name='2014height',
                                clump_file_name='2014height',
                                qc_file_name='2014height',
                                clump_kb='250',
                                clump_p1='0.02',
                                clump_r2='0.5',clumpfolder_name='2014height')
```

- new snps list obtained

`result/plink/clump/*.qc_clump_snpslist.csv`

12.10 Step4.1 Generate PRS model

In step4 the function `build-prs` is built on Plink2.0 dosage.

```
$ gprs build-prs --vcf_input [str] --output_name [str] --qc_clump_snplist_foldername␣
↪[str] --memory [int] --clump_kb [int] --clump_p1 [float/scientific notation] --clump_
↪r2 [float] --symbol [str/int] --columns [int] --plink_modifier [str]
```

- In real use:

```
$ gprs build-prs --vcf_input 1000genomes/hg19 --symbol . --qc_file_name 2014height --
↪columns 1 2 3 --memory 1000 --clump_kb 250 --clump_p1 0.02 --clump_r2 0.1 --output_
↪name 2014height
```

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.build_prs( vcf_input= 'home/1000genomes/hg19',
                         output_name ='2014height', memory='1000',clump_kb='250',
                         clump_p1='0.02', clump_r2='0.02', qc_clump_snplist_foldername=
↪'2014height')
```

- chr1-chr22 sscore files obtained

`result/plink/prs/*.sscore`

12.11 Step4.2 Combined PRS model

From step 4.1 we will have 1-22 chromosomes `sscore` files, in this step we are going to combine these files as one output.

```
$ gprs build-prs --vcf_input [str] --symbol [str/int] --qc_file_name [str] --columns␣
↪[int] --plink_modifier [str] --memory [int] --clump_kb [int] --clump_p1 [float/
↪scientific notation] --clump_r2 [float] --output_name [output name] --clump_kb [int] --
↪clump_p1 [float/scientific notation] --clump_r2 [float]
```

- In real use:

```
$ gprs build-prs --vcf_input 1000genomes/hg19 --symbol . --qc_file_name 2014height --
↪columns 1 2 3 --memory 1000 --clump_kb 250 --clump_p1 0.02 --clump_r2 0.1 --output_
↪name 2014height
```

```
from gprs.gene_atlas_model import GeneAtlasModel

if __name__ == '__main__':
    geneatlas = GeneAtlasModel( ref='1000genomes/hg19',
                    data_dir='data/2014_GWAS_Height' )

    geneatlas.combine_prs(filename="2014height",
                          clump_r2="0.5",clump_kb="250",clump_p1="0.02")
```

- one combined sscore files obtained

`result/plink/prs/*.sscore`

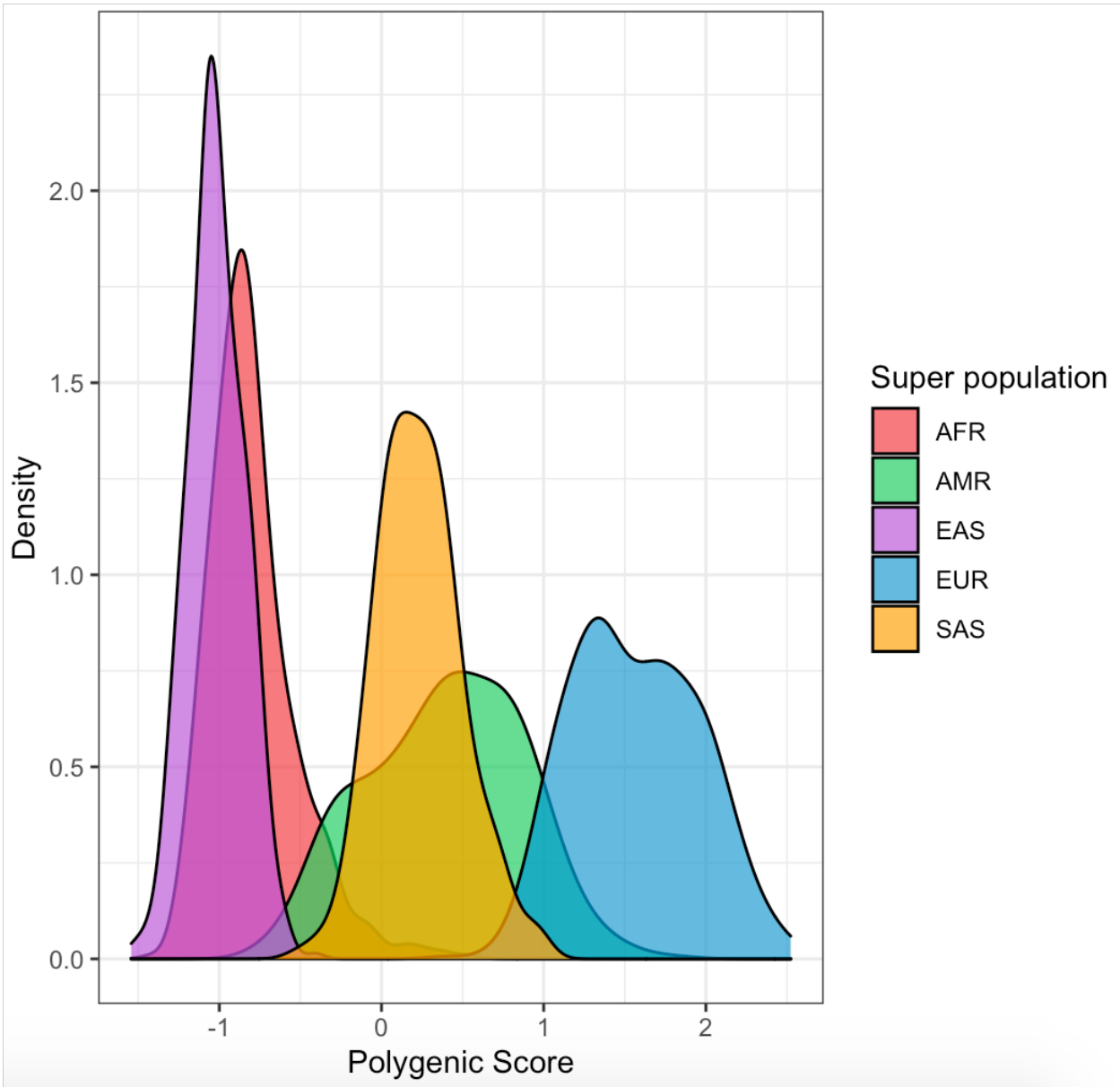
12.12 Visualize distribution of PRS in each population

```
# Libraries
library(ggplot2)
library(dplyr)

data <- read.csv("combine_profil_w_pop.txt", sep=" ")

data$SCORE_Z <- (data$SCORE-mean(data$SCORE))/sd(data$SCORE)

ggplot(data, aes(x=data$SCORE_Z, group=data$super_pop, fill=data$super_pop)) +
  geom_density(adjust=1.25, alpha=.7) +
  scale_fill_manual(values=c("brown2","springgreen3","mediumorchid","deepskyblue3",
↪"orange"))+
  labs(x = "Polygenic Score", y = "Density", fill = "Super population")+
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```


Density
2.0
1.5
1.0
0.5
0.0
-1
0
1
2
Polygenic Score
Super population
AFR
AMR
EAS
EUR
SAS

CHAPTER

THIRTEEN

EXAMPLE OF MEC AMERICAN AFRICANS

This is an example of MEC American Africans

Under construction

CHAPTER

FOURTEEN

GPRS

CHAPTER

FIFTEEN

GENEATLASMODEL

CHAPTER

SIXTEEN

GWASMODEL

CHAPTER

SEVENTEEN

INDICES AND TABLES

- genindex
- modindex
- search