This document is set up to follow all steps to include a new GWAS in the PRSMultiTrait package. ***Concise instructions are in the later pages of this document.***

Prep:

I’m using the BMI from Yengo et al GWAS as an example. This is a realistic example,

I open the GWAS, check what columns exist.

In this case its: CHR BP SNP A1 A2 Freq\_Tested\_Allele\_in\_HRS BETA SE P N

Chr and BP indicate the pos in chr:bp

SNP is rsID

A1 is tested allele

A2 is other allele

Freq\_Tested\_Allele\_in\_HRS, some specific allele frequency

SE standard error in calculating beta

P is the significance

Beta is the direction

N is number of samples

To get this into correct format for input for the curated GWAS it need:

Colnames: CHR BP A1 A2 N BETA/OR P

Chr: chromosome in GRCh37

Bp: basepair location in GRCh37

A1: effect allele

A2: other allele

N: number of samples

Beta or OR: Beta coef or odds ratio

P is the significance

Then this file is saved as a tab-separated file with decimal separator as dot, thousands separator as comma. [short trait abbreviation]\_[year in yyyy]\_[1st author]\_curated.summaries.

eg: BMI\_2018\_Yengo\_curated.summaries

save the file somewhere. (In my case D:\DATA\_STORAGE\GWAS\GWAS\Curated\_GWASes\BMI\_2018\_Yengo\_curated.summaries)

Manifest manipulation

This GWAS information needs to be added to the manifest before anything else.

I call the library PRSMultiTrait and get the manifest.

library(PRSMultiTrait)

PRSMultiTrait::getManifest()

Then I use the function to add a new GWAS to the manifest ‘addGWAStoManifest()’. This function has all the fields to be entered in the manifest the defaults are:

short = c("UniqueTraitName"), # UNIQUE name to describe trait, further used to select this gwas

n = c(10000), # enter the sample size, as accurate as possible, but this scales in orders of magnitude

filename = c("?"), # the EXACT PATH where the summaries file is located (eg: D:/DATA\_STORAGE/GWAS/GWAS/Curated\_GWASes/BMI\_2018\_Yengo\_curated.summaries),

year = c("?"), # year, for pheno only,

trait = c("?"), # a short description of the trait included

DOI = c("?"), # doi, for pheno only,

genomeBuild = c("?"), # build, for pheno only, should be 37

traitType = c("?"), # either EXACT “CAT” or “CONT”. categorical / continuous

rawSNPs = c("?"), # leave it as is

finalModelSNPs = c("?"), # leave it as is

modelRunningTime = c("?"), # leave it as is

usedRefSet = c("?"), # not used yet

processed = c(0), # leave 0

FORCE = FALSE) # used to omit the confirmation message, set to true to don’t get message

PRSMultiTrait::addGWAStoManifest(short = c("BMI"), n = c(689928), filename = c("D:/DATA\_STORAGE/GWAS/GWAS/Curated\_GWASes/BMI\_2018\_Yengo\_curated.summaries"),

year = c("2018"), trait = c("Body mass index"), DOI = c("some link"), genomeBuild = c("GRCh37"),

traitType = c("CONT"), rawSNPs = c("?"), finalModelSNPs = c("?"),

modelRunningTime = c("?"), usedRefSet = c("?"), processed = c(0),

FORCE = TRUE)

Automated GWAS preparation

If everything is entered correctly, the GWAS can be loaded in the package and processed to be included in its ‘internal’ database. This is a more processed version of the summaries file, respectively. This can be done by calling the “prepareGWAS()” function. It requires only the short trait ID to work.

PRSMultiTrait::prepareGWAS(trait = "BMI")

Should give a ‘Done processing BMI!’ check at the end.

Generation of PGM

The next step is to calculate the polygenic model using the ‘calcPGS\_LDAK’ function. This function takes two arguments, the short trait ID and the method (for now fixed at ‘bayesr’).

PRSMultiTrait::calcPGS\_LDAK = function(Trait = "BMI",Model = "bayesr")

That should run for a couple of minutes and should give a ‘Completed’ check at the end (along with some other information where it is stored)

That should be it! A new GWAS is now included in its internal database, and can be used in predicting PGS for other cohorts.

Concise instructions:

Prep:

1. Open the GWAS and check the columns.
2. Make sure the columns match the following format: CHR, BP, A1, A2, N, BETA/OR, P.
3. Save the file as a tab-separated file with decimal separator as dot, thousands separator as comma with the following naming convention: [short trait abbreviation]\_[year in yyyy]\_[1st author]\_curated.summaries. (e.g. BMI\_2018\_Yengo\_curated.summaries)
4. Save the file in a desired location (e.g. D:\DATA\_STORAGE\GWAS\GWAS\Curated\_GWASes\BMI\_2018\_Yengo\_curated.summaries).

Manifest manipulation:

1. Load the PRSMultiTrait library: library(PRSMultiTrait)
2. Get the current manifest using the function PRSMultiTrait::getManifest()
3. Use the function addGWAStoManifest() to add the new GWAS to the manifest.

Automated GWAS preparation:

1. Use the function prepareGWAS() and provide the trait ID as the input. For example: PRSMultiTrait::prepareGWAS(trait = "BMI")

Generation of PGM:

1. Use the function calcPGS\_LDAK() and provide the trait ID as the input and set the method (bayesr for now). For example: PRSMultiTrait::calcPGS\_LDAK = function(Trait = "BMI",Model = "bayesr")