A real example for performing GWAS summary-level data based MR analysis with MRAPSS package

HU Xianghong

10/09/2020

Introduction

The MR-APSS is a unified approach to Mendelian Randomization accounting for pleiotropy and sample structure using genome-wide summary statistics. Specifically, MR-APSS uses a background-foreground model to characterize the estimated effects of SNPs on exposure and outcome, where the background model accounts for confounding from pleiotropy and sample structure, and the foreground model captures the valid signal for causal inference.

We illustrate how to analyze GWAS summary-level data using the MR-APSS software by a real example of BMI (exposure) and T2D (outcome). The MR-APSS model analysis comprises the following steps:

- Step 1: Prepare data and estimate nuisance parameters
- Step 2: Fit MR-APSS for causal inference

Step 1 requires the LD score files at link. You can also use the LD scores calculated by yourself.

Step 0: Installation and load packages

```
#install.packages("devtools")
devtools::install_github("YangLabHKUST/MR-APSS")
```

```
library(MRAPSS)
library(readr)
```

Step 1: Prepare data and estimate nuisance parameters

1.1. Download GWAS summary-level data from public resources

To begin, we should set your working directory use setwd(). We download the the GWAS summary statistics at links for BMI(Watanabe et.al. (2019) [PMID: 31427789]) and T2D(Xue et al. (2015) [PMID: 30054458]). We then read the downloaded datasets into R which are renamed as "BMI_ukb.txt" and "T2D.txt"

```
BMI_raw <- readr::read_delim("BMI_ukb.txt", "\t", escape_double = FALSE,
                         trim_ws = TRUE, progress = F)
#> Parsed with column specification:
#> cols(
#>   .default = col_double(),
#>   SNP = col_character(),
#>   A1 = col_character(),
#>   TEST = col_character(),
#>   A2 = col_character(),
#>   SNPID_UKB = col_character(),
#>   A1_UKB = col_character(),
#>   A2_UKB = col_character()
#> )
```

```
#> See spec(...) for full column specifications.

T2D_raw <- readr::read_delim("T2D.txt", " ",
                             escape_double = FALSE,
                             trim_ws = TRUE, progress = F)
#> Parsed with column specification:
#> cols(
#>   CHR = col_double(),
#>   BP = col_double(),
#>   SNP = col_character(),
#>   A1 = col_character(),
#>   A2 = col_character(),
#>   frq_A1 = col_double(),
#>   b = col_double(),
#>   se = col_double(),
#>   P = col_double(),
#>   N = col_double()
#> )
```

1.2. Format summary statistics

Following LDSC, we format the summary-level data to have the following columns by format_data():

- SNP: rs number
- A1: effect allele
- A2: the other allele
- Z: z score
- chi2: χ^2 statistics
- P: p-value
- N: sample size

MR-APSS needs the following information from each GWAS data set:

- rs number
- effect allele
- the other allele
- sample size
- a signed summary statistic (anything that can be converted to a z-score)

The format_data() function will try to interpretate the raw datsets with user specified column names. You can set rs number, effect allele and the other allele using “snp_col”,“A1_col” and “A2_col” arguements. A signed summary statistic can be obtained by using one or two of the following columns: “b_col”, “or_col”, “z_col”,“se_col”, “p_col” (p-value). You can set sample size using “n_col”, or “ncase_col” and "“ncontrol_col. If the column for sample size is not available, you can use argument”n" to specify the total sample size. It could be nice if MAF (“freq_col”) and INFO (“info_col”) are available.

The format_data() function will also conduct the following quality control procedures.

- extract SNPs in the set of HapMap 3 list
- remove SMPs with minor allele frequency < 0.05 (if freq_col colum is avaliable).

- remove SNPs with alleles not in (G,C,T,A).
- remove SNPs with ambiguous alleles (G/C or A/T) or other false alleles (A/A T/T, G/G or C/C).
- remove SNPs with Info < 0.9 (if info_col column is available).
- exclud SNPs in the complex Major Histocompatibility Region (Chromosome 6, 26Mb-34Mb).
- remove SNPs with $\chi^2 > \chi 2_{max}$. The default value for $\chi 2_{max}$ is $\max(N/1000, 80)$.

Now, we use format_data() to obtain the formatted data sets for BMI and T2D.

```
BMI = format_data(BMI_raw,
                  snp_col = "SNPID_UKB",
                  b_col = "BETA",
                  se_col = "SE",
                  freq_col = "MAF_UKB",
                  A1_col = "A1",
                  A2_col = "A2",
                  p_col = "P",
                  n_col = "NMISS",
                  info_col = "INFO_UKB")
#> Begin formatting ....
#> The raw dataset has 10599054 dat lines
#> Remove SNPs with imputation info less than 0.9 ..., remaining 10599018 SNPs.
#> Remove ambiguous SNPs ..., remaining 8995068 SNPs.
#> Remove SNPs in MHC region ..., remaining 8944044 SNPs.
#> Remove  duplicated SNPs ..., remaining 8944044 SNPs.
#> Merge SNPs with the hapmap3 snplist ..., remaining 1161501 SNPs.
#> Remove SNPs with alleles not matched with the hapmap3 snplist, remaining 1161195 SNPs.
#> Remove SNPs with p value < 0 or p value > 1, remaining 1037824 SNPs.
#> Infer z score from p value and b ...
#> Remove missing values, remaining 1037824 SNPs.
#> Remove SNPs with sample size 5 standard deviations away from the mean, remaining 1037824 SNPs.
#> Remove SNPs with chi2 > chi2_max ... , remaining 1037799 SNPs.
#> The formatted data has 1037799 dat lines.

T2D = format_data(T2D_raw,
                  snp_col = "SNP",
                  b_col = "b",
                  se_col = "se",
                  freq_col = "frq_A1",
                  A1_col = "A1",
                  A2_col = "A2",
                  p_col = "P",
                  n_col = "N")
#> Begin formatting ....
#> The raw dataset has 5053015 dat lines
#> Remove ambiguous SNPs ..., remaining 4280791 SNPs.
#> Remove SNPs in MHC region ..., remaining 4274525 SNPs.
#> Remove  duplicated SNPs ..., remaining 4274525 SNPs.
#> Merge SNPs with the hapmap3 snplist ..., remaining 1007638 SNPs.
#> Remove SNPs with p value < 0 or p value > 1, remaining 936658 SNPs.
#> Infer z score from p value and b ...
#> Remove missing values, remaining 936658 SNPs.
#> Remove SNPs with sample size 5 standard deviations away from the mean, remaining 936565 SNPs.
#> Remove SNPs with chi2 > chi2_max ... , remaining 936556 SNPs.
```

```
#> The formatted data has 936556 dat lines.
```

We can have a look at the formatted datasets:

```
head(BMI)
#>          SNP A1 A2          Z      N       chi2       P
#> 1  rs1000000  A  G -0.2879304 385270 0.08290393 0.77340
#> 2 rs10000010  C  T -0.7299839 376552 0.53287656 0.46540
#> 3  rs1000002  T  C -1.9961279 385336 3.98452670 0.04592
#> 4 rs10000023  G  T  2.5662659 371893 6.58572050 0.01028
#> 5  rs1000003  G  A -1.5813414 383709 2.50064060 0.11380
#> 6 rs10000033  C  T  1.4286669 382629 2.04108907 0.15310
```

```
head(T2D)
#>          SNP A1 A2          Z      N       chi2       P
#> 1  rs1000000  A  G -0.2881917 573633 0.08305446 0.77320
#> 2 rs10000010  C  T  0.5018117 587226 0.25181496 0.61580
#> 3  rs1000002  T  C -2.5639113 579347 6.57364100 0.01035
#> 4 rs10000023  G  T  0.3003634 557936 0.09021815 0.76390
#> 5  rs1000003  G  A -1.6381040 577450 2.68338473 0.10140
#> 6 rs10000033  C  T  1.9173774 576206 3.67633619 0.05519
```

1.3. Harmonize the formatted datasets and estimate nuisance parameters

This step is designed for merging and harmonizing the formatted datasets from step 1.2 to make sure effect sizes for the same SNP correspond to the same allele for the exposure and outcome, and then, estimating the variance-covariance matrix $\mathbf{\Omega}$ and $\mathbf{C}$ in the background model by LD score regression.

The analysis for this step is accomplished by function est_paras():

```
paras = est_paras(dat1 = BMI,
                  dat2 = T2D,
                  trait1.name = "BMI",
                  trait2.name = "T2D",
                  ldscore.dir = "./eur_w_ld_chr")
#> Merge dat1 and dat2 by SNP ...
#> Harmonise the direction of SNP effects of exposure and outcome
#> Read in LD scores ...
#> Add LD scores to the harmonised data sets...
#> The Harmonised dataset will also be used for  MR analysis
#> Begin estimation of C and Omega using LDSC ...
#> Estimate heritability for trait 1 ...
#> Using two-step estimator with cutoff at 30.
#> Mean Chi2:3.0256.
#> Intercept: 1.1521(0.025).
#> Total Observed scale h2:0.2174(0.0071).
#> Estimate heritability for trait 2 ...
#> Using two-step estimator with cutoff at 30.
#> Mean Chi2:1.7045.
#> Intercept: 1.0911 (0.0167).
#> Total Observed scale h2:0.0485 (0.0028).
#> Estimate genetic covariance ...
#> Using two-step estimator with cutoff at 30.
#> Intercept: 0.1618 (0.0145).
#> Total Observed scale gencov: 0.054 (0.003).
```

Here, the argument "ldscore.dir" specifies the path to LD score files.

Now, we can try the following commands to check the estimates:

```
paras$Omega
#>               [,1]         [,2]
#> [1,] 1.852220e-07 4.599093e-08
#> [2,] 4.599093e-08 4.132106e-08
```

```
paras$C
#>           [,1]      [,2]
#> [1,] 1.1520504 0.1617687
#> [2,] 0.1617687 1.0911428
```

Note that the non-diagonal elements of **C** is the intercept estimate of cross-trait LD score regression; the diagonal elements of **C** are the intercept estimates of single-trait LD score regressions. The diagonal elements can be fixed at 1 by specify "h2.intercept = T" in function "est_params()".

The harmonized dataset will be used for LD clumping at the next stage.

```
head(paras$dat)
#>          SNP A1 A2         b.exp         b.out      se.exp      se.out
#> 1  rs1000000  A  G -0.0004638793 -0.0003805085 0.001611081 0.001320331
#> 2 rs10000010  C  T -0.0011895996  0.0006548444 0.001629624 0.001304960
#> 3  rs1000002  T  C -0.0032156486 -0.0033684772 0.001610943 0.001313804
#> 4 rs10000023  G  T  0.0042081639  0.0004021192 0.001639800 0.001338776
#> 5  rs1000003  G  A -0.0025528462 -0.0021556799 0.001614355 0.001315960
#> 6 rs10000033  C  T  0.0023096280  0.0025259149 0.001616632 0.001317380
#>   pval.exp pval.out       L2
#> 1  0.77340  0.77320 20.90684
#> 2  0.46540  0.61580 23.52035
#> 3  0.04592  0.01035 46.00095
#> 4  0.01028  0.76390 25.62940
#> 5  0.11380  0.10140 37.94950
#> 6  0.15310  0.05519 10.62294
```

1.4. IV selection and LD clumping

IV selection and LD clumping are accomplished by function 'clump()'. The default IV threshold for MR-APSS is 5×10^{-5}. One can specify the IV selection threshold to obtain a selected dataset satisfying "pval.exp $<$ Threshold". The plink clumping procedure is used for LD clumping to extract a data frame for a set of nearly independent SNPs from the selected SNPs.

```
MRdat = clump(paras$dat,
              IV.Threshold = 5e-05,
              SNP_col = "SNP",
              pval_col = "pval.exp",
              clump_kb = 1000,
              clump_r2 = 0.001)
#> API: public: http://gwas-api.mrcieu.ac.uk/
#> Please look at vignettes for options on running this locally if you need to run many instances of th
#> Clumping RSO5yC, 26527 variants, using EUR population reference
#> Removing 25300 of 26527 variants due to LD with other variants or absence from LD reference panel
```

By default, clump() performs LD clumping through API, which means that PLINK is not required in your machine (see dependencies in https://github.com/MRCIEU/ieugwasr). Of course, one can perform LD clumping locally using PLINK by specifying the LD reference panel with "bfiles" and the path to local plink

binary with"plink_bin" (see ?clump()).

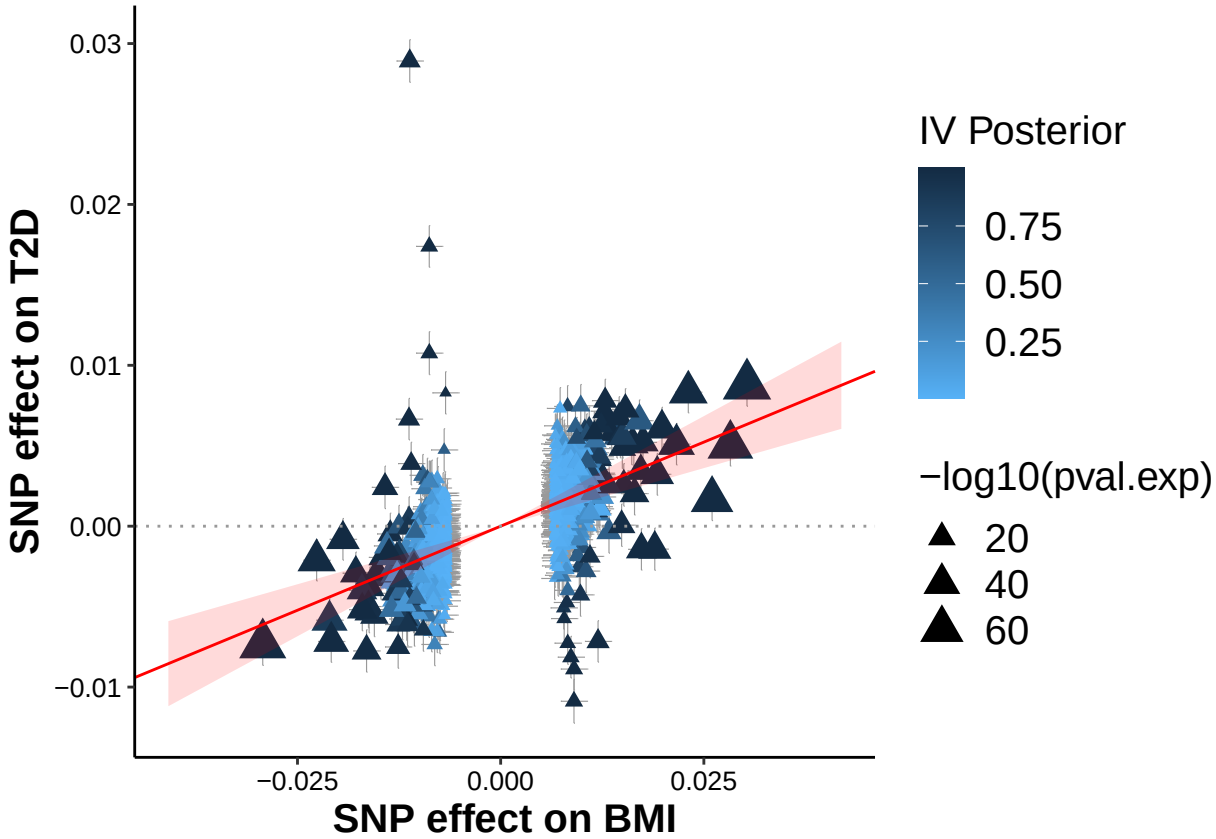
Step 2: Fit MR-APSS for causal inference

Now we can fit the MR-APSS model with the selected IVs (MRdat) and the pre-estimated background parameters, i.e., $\mathbf{\Omega}$ and $\mathbf{C}$.

```
MRres = MRAPSS(MRdat,
               exposure = "BMI",
               outcome = "T2D",
               C = paras$C,
               Omega = paras$Omega ,
               Cor.SelectionBias = T)
#> ************************************************************
#> MR test results of  BMI  on  T2D :
#> MR-APSS: beta =  0.209 beta.se =  0.033 pvalue =  2.3976e-10 #SNPs=  1227
#> # valid IVs with foreground signals:  178.019
#> ************************************************************
```

We can visualize MR-APSS analysis results by "MRplot()":

```
MRplot(MRres, exposure = "BMI", outcome = "T2D")
```



The figure shows the results of MR-APSS using the default IV threshold 5×10^{-5}. The estimated causal effect is indicated by a red line with its 95% confidence interval indicated by the shaded area in transparent red color. Triangles indicate the observed SNP effect sizes (γ_j and Γ_j). The color of triangles indicates the posterior of a valid IV, i.e., the posterior of an IV carrying the foreground signal ($Z_j = 1$, dark blue) or not ($Z_j = 0$, light blue).