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

HU Xianghong

10/09/2020

Introduction

MR-APSS is a unified approach for Mendelian randomization accounting for pleiotropy, sample overlap and selection bias using genome-wide summary statistics. The MRAPSS package implement the MR-APSS approach to test for the causal effects between an exposure and a outcome disease.

We illustrate how to analyze GWAS summary level data using the MRAPSS software by an real example, i.e. BMI (exposure) and T2D (outcome). The MRAPSS analysis comprises six steps:

- Step 1: Download GWAS summary-level data from public resources
- Step 2: Format data
- Step 3: Harmonise datasets and estimate nuisance parameters
- Step 4: IVs selection and LD clumping
- Step 5: Fit MRAPSS
- Step 6: Visualize

Step 3 requires the LD score files at link. You can also use the LD scores calculated by yourself.

Step 0: Installition and load packages

```
#install.packages("devtools")
devtools::install_github("YangLabHKUST/MRAPSS")
```

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

Step 1: Download GWAS summary-level data from public resources

To begin, set your working directory use setwd().

Download the GWAS summary statistics at links for BMI(Watanabe et.al. (2019) [PMID: 31427789]) and T2D(Xue et al. (2015) [PMID: 30054458]).

You can also download the datasets here

Uncompress and rename the files as "BMI_ukb.txt" and "T2D.txt", then read the datasets into R:

```
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()
#> )
```

Step 2: Format summary statistics

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: pvalue
- N: sample size

```
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.
#> Removing SNPs with imputation info less than 0.9 ...
#> Removing ambiguous SNPs ...
#> Removing SNPs in MHC region ...
#> Removing  duplicated SNPs ...
#> Merge SNPs with the hapmap3 snplist ...
#> Removing SNPs with alleles not matched with the hapmap3 snplist
#> Removing SNPs with p value < 0 or p value > 1
#> Infer z score from p value and b ...
#> Removing SNPs with sample size 5 standard deviations away from the mean
#> Removing SNPs with chi2 > chi2_max ...
#> 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.
#> Removing ambiguous SNPs ...
#> Removing SNPs in MHC region ...
#> Removing  duplicated SNPs ...
#> Merge SNPs with the hapmap3 snplist ...
#> Removing SNPs with p value < 0 or p value > 1
#> Infer z score from p value and b ...
#> Removing SNPs with sample size 5 standard deviations away from the mean
#> Removing SNPs with chi2 > chi2_max ...
#> The formatted data has 936556 dat lines.
```

Have a look at the formmated 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
```

Note: format_data() will try to interpretate the raw datsets with user specified column names, for example specify “SNP” in “BMI.txt” as “SNP_col”. At the same time, it will also conduct the following quality control procedures.

- extract SNPs in the set of HapMap 3 list with minor allele frequency > 0.05.
- 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 orC/C).
- exclud SNPs in the complex Major Histocompatibility Region (Chromosome 6, 26Mb-34Mb).
- remove SNPs with $\chi^2 > chi2_{max}$. The default value for $chi2_{max}$ is $\max(N/1000, 80)$.

Step 3: Harmonise the formmated datasets and estimate nuisance parameters

This step is desined for mergeing and harmonizing the formatted datasets from step 2 to make sure effect sizes for the same SNP correspond to the same allele for the exposure and outcome.

Then estimate the variance-covariance matrix $\mathbf{\Omega}$ in background model and the residual correlation parameter ρ due to sample overlap through implemetation of LD score regression.

The analysis for this step can be 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 trait 1 and trait 2
#> 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 Sigma 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).
#> Time elapsed: 3515.322
```

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

Try the following commands to see the estimates:

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

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

Note: the non-diagonal elements of Sigma_err is the estimate of ρ by the intercept of cross-trait LD score regression. The diagnonal elements of Sigma_err is the intercept of single-trait LD score regressions. The intercept can be fixed at 1 by specify "h2.intercept = T" in function "est_params()".

The harmonized dataset will be used for LD clumping.

```
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
```

Step 4: IVs selection and LD clumping

This step is accomplished by function 'clump()'. One should specify the IV selection threshold to obtain a selected dataset satisfying "pval.exp < Threshold". The default IV threshold for MR-APSS is 5×10^{-8}. The plink clumping prosedure is used for LD clumping to extract a data frame for a set of nerarly 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 XOEvqY, 26527 variants, using EUR population reference
#> Removing 25300 of 26527 variants due to LD with other variants or absence from LD reference panel
```

Note: by default, clump() pefroms LD clumping through API, which means you don't need to install PLINK tools in your machine (see depencies in https://github.com/MRCIEU/ieugwasr). For sure, you can do LD clumping locally with PLINK, (see ?clump()).

Step 5: Fit MRAPSS

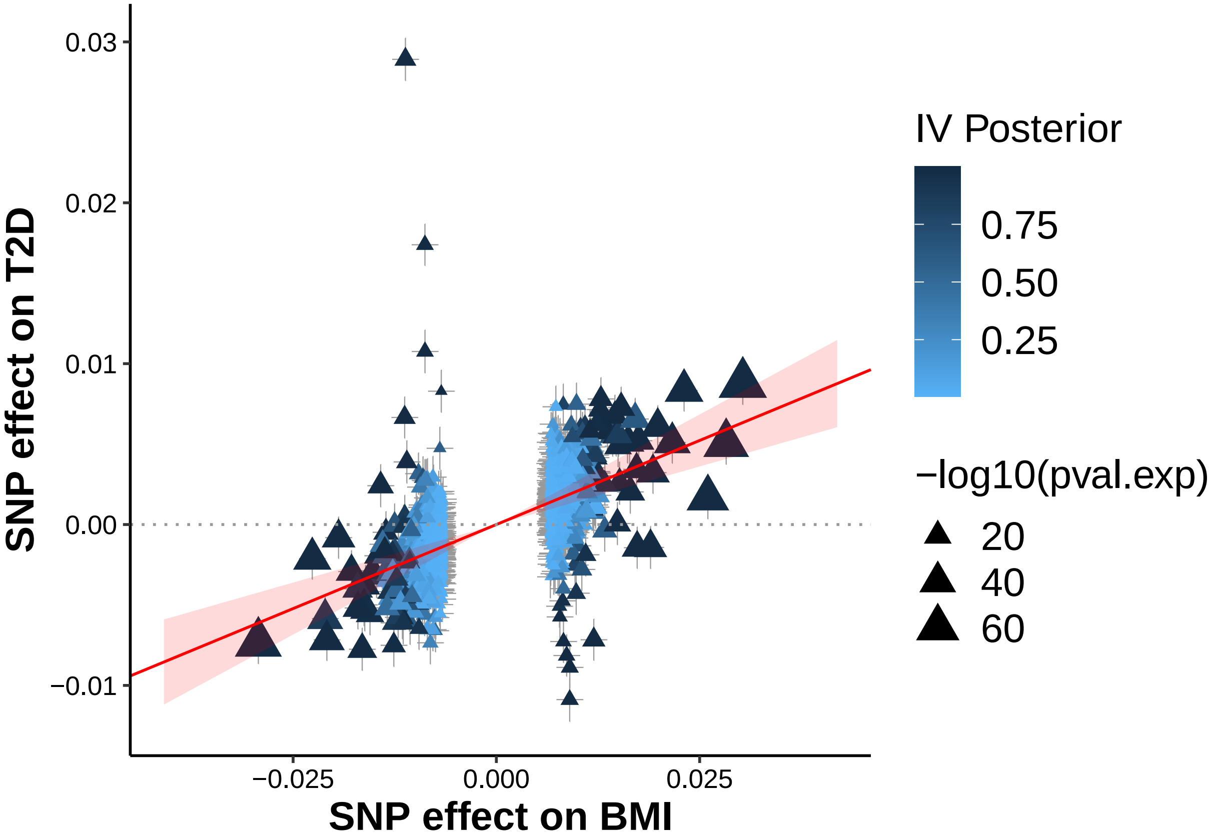
Fit MRAPSS when parameters estimates (Sigma_err and Omega) and summary statistics of clumped SNPs (MRdat) are ready.

```r
MRres = MRAPSS(MRdat,
               exposure = "BMI",
               outcome = "T2D",
               Sigma_err = paras$Sigma_err,
               Omega = paras$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
#> Correlation parameter (rho) due to sample overlap :  0.1617687
#> Proportion of effective IVs with foreground signals:  0.1450847
#> Variance component (Omega) for background model =
#>              [,1]         [,2]
#> [1,] 1.852220e-07 4.599093e-08
#> [2,] 4.599093e-08 4.132106e-08
#> Variance component (Sigma) for foreground model =
#>              [,1]        [,2]
#> [1,] 2.731423e-06 0.00000e+00
#> [2,] 0.000000e+00 7.27557e-07
#> ***********************************************************
```

Step 6: Visualize

Visualize MRAPSS analysis results by “MRplot()”:

```r
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).