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

HU Xianghong

5/8/2020

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. LDL-C (exposure) and CAD(outcome). The MRAPSS analysis comprises five 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 3 requires the LD score files at link. You can also use the LD scores calculated by yourself.

You can also try reverse direction MR analysis (CAD -> LDL-C) based on the output of step 3.

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 LDL-C(Willer et al (2013) [PMID: 24097068]) and CAD(Nikpay et al. (2015) [PMID: 26343387]).

You can also download the datasets here

Uncompress and rename the files as “LDL-C.txt” and “CAD.txt”, then read the datasets into R:

```
LDL_raw <- readr::read_delim("LDL-C.txt", "\t", escape_double = FALSE,
                             trim_ws = TRUE, progress = F)
#> Parsed with column specification:
#> cols(
#>   SNP_hg18 = col_character(),
#>   SNP_hg19 = col_character(),
#>   rsid = col_character(),
#>   A1 = col_character(),
#>   A2 = col_character(),
#>   beta = col_double(),
#>   se = col_double(),
#>   N = col_double(),
#>   `P-value` = col_double(),
#>   Freq.A1.1000G.EUR = col_double()
#> )

CAD_raw <- readr::read_delim("CAD.txt", "\t", escape_double = FALSE,
                          trim_ws = TRUE, progress = F)
#> Parsed with column specification:
#> cols(
#>   markername = col_character(),
#>   chr = col_double(),
#>   bp_hg19 = col_double(),
#>   effect_allele = col_character(),
#>   noneffect_allele = col_character(),
#>   effect_allele_freq = col_double(),
#>   median_info = col_double(),
#>   model = col_character(),
#>   beta = col_double(),
#>   se_dgc = col_double(),
#>   p_dgc = col_double(),
#>   het_pvalue = col_double(),
#>   n_studies = 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

```
LDL = format_data(LDL_raw,
                  snp_col = "rsid",
                  b_col = "beta",
                  se_col = "se",
                  freq_col = "Freq.A1.1000G.EUR",
                  A1_col = "A1",
                  A2_col = "A2",
                  p_col = "P-value",
                  n_col = "N")
```

```
#> Begin formatting ....
#> The raw dataset has 2437751 dat lines.
#> Remove SNPs in MHC region ...
#> Merge SNPs with the hapmap3 snplist ...
#> Remove ambiguous SNPs ...
#> Remove SNPs with alleles not matched with the hapmap3 snplist
#> Remove SNPs with p value < 0 or p value > 1
#> Infer Z score from b/se ...
#> Remove SNPs with chi2 > max(n/1000,80)...
#> The formatted data has 1037423 dat lines.

CAD = format_data(CAD_raw,
                  snp_col = "markername",
                  b_col = "beta",
                  se_col = "se_dgc",
                  freq_col = "effect_allele_freq",
                  A1_col = "effect_allele",
                  A2_col = "noneffect_allele",
                  p_col = "p_dgc",
                  n = 185000,
                  info_col = "median_info")
#> Begin formatting ....
#> The raw dataset has 9455778 dat lines.
#> Remove SNPs in MHC region ...
#> Merge SNPs with the hapmap3 snplist ...
#> Remove SNPs with imputation info less than 0.9 ...
#> Generating sample size from specified sample size
#> Remove SNPs with p value < 0 or p value > 1
#> Infer Z score from b/se ...
#> Remove SNPs with chi2 > max(n/1000,80)...
#> The formatted data has 1129971 dat lines.
```

Have a look at the formmated datasets:

```
head(LDL)
#>          SNP A1 A2         Z      N       chi2       P
#> 1  rs1000000  A  G 0.8064516  89888 0.65036420 0.51210
#> 2 rs10000010  T  C 1.6111111 173002 2.59567901 0.13150
#> 3  rs1000002  C  T 0.9423077  89785 0.88794379 0.58690
#> 4 rs10000023  G  T 0.2264151  89888 0.05126379 0.70160
#> 5  rs1000003  G  A 1.4057971  89734 1.97626549 0.20900
#> 6 rs10000033  T  C 1.7500000  89867 3.06250000 0.09375
```

```
head(CAD)
#>          SNP A1 A2           Z      N         chi2         P
#> 1  rs1000000  G  A -0.98523627 185000 0.9706905146 0.3245096
#> 2 rs10000010  C  T -0.01741633 185000 0.0003033284 0.9861045
#> 3  rs1000002  C  T  0.71910849 185000 0.5171170190 0.4720730
#> 4 rs10000023  T  G -0.74076465 185000 0.5487322716 0.4588365
#> 5  rs1000003  A  G -0.74388130 185000 0.5533593939 0.4569476
#> 6 rs10000033  T  C  0.48246723 185000 0.2327746294 0.6294745
```

Note: format_data() will try to interpretate the raw datsets with user specified column names, for example specify "rsid" in LDL.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>0.01.
- 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 $\chi^2 > \max(80, n/1000)$.

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 = LDL,
                  dat2 = CAD,
                  trait1.name = "LDL-C",
                  trait2.name = "CAD",
                  ldscore.dir = "./eur_w_ld_chr")
#> Merge data1 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 Chi2:1.2049.
#> Intercept: 0.9562(0.0091).
#> Total Observed scale h2:0.1471(0.0119).
#> Estimate heritability for trait 2 ...
#> Using two-step estimator with cutoff at 30.
#> Mean Chi2:1.123.
#> Intercept: 0.8855 (0.0083).
#> Total Observed scale h2:0.0638 (0.005).
#> Estimate genetic covariance ...
#> Using two-step estimator with cutoff at 30.
#> Intercept: 0.0148 (0.0055).
#> Total Observed scale gencov: 0.0211 (0.0042).
#> Time elapsed: 167.697
```

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,] 1.253563e-07 1.798647e-08
#> [2,] 1.798647e-08 5.433895e-08
```

```
paras$Sigma_err
#>            [,1]       [,2]
#> [1,] 0.95618182 0.01479736
#> [2,] 0.01479736 0.88553510
```

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 0.002689846 2.290628e-03 0.003335409 0.002324953
#> 2 rs10000010  T  C 0.003873468 4.049214e-05 0.002404221 0.002324953
#> 3  rs1000002  C  T 0.003144784 1.671893e-03 0.003337322 0.002324953
#> 4 rs10000023  G  T 0.000755187 1.722243e-03 0.003335409 0.002324953
#> 5  rs1000003  G  A 0.004692931 1.729489e-03 0.003338270 0.002324953
#> 6 rs10000033  T  C 0.005837648 1.121714e-03 0.003335799 0.002324953
#>   pval.exp  pval.out       L2
#> 1  0.51210 0.3245096 20.90684
#> 2  0.13150 0.9861045 23.52035
#> 3  0.58690 0.4720730 46.00095
#> 4  0.70160 0.4588365 25.62940
#> 5  0.20900 0.4569476 37.94950
#> 6  0.09375 0.6294745 10.62294
```

Step 4: IVs selection and LD clumping

Specify the IV selection threshold to obtain a selected dataset satisfying "pval.exp < Threshold". Then apply the "clump()" function to the selected SNPs, which uses the PLINK clump method to extract a data frame for a set of nerarly independent SNPs.

```
Threshold = 5e-05       # IV selection threshold
MRdat = clump(subset(paras$dat, pval.exp < Threshold),
              SNP_col = "SNP",
              pval_col = "pval.exp",
              clump_kb = 1000,
              clump_r2 = 0.001)
#> API: public: http://gwas-api.mrcieu.ac.uk/
#> Clumping j8Iqe7, 2292 variants
#> Removing 2115 of 2292 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()).

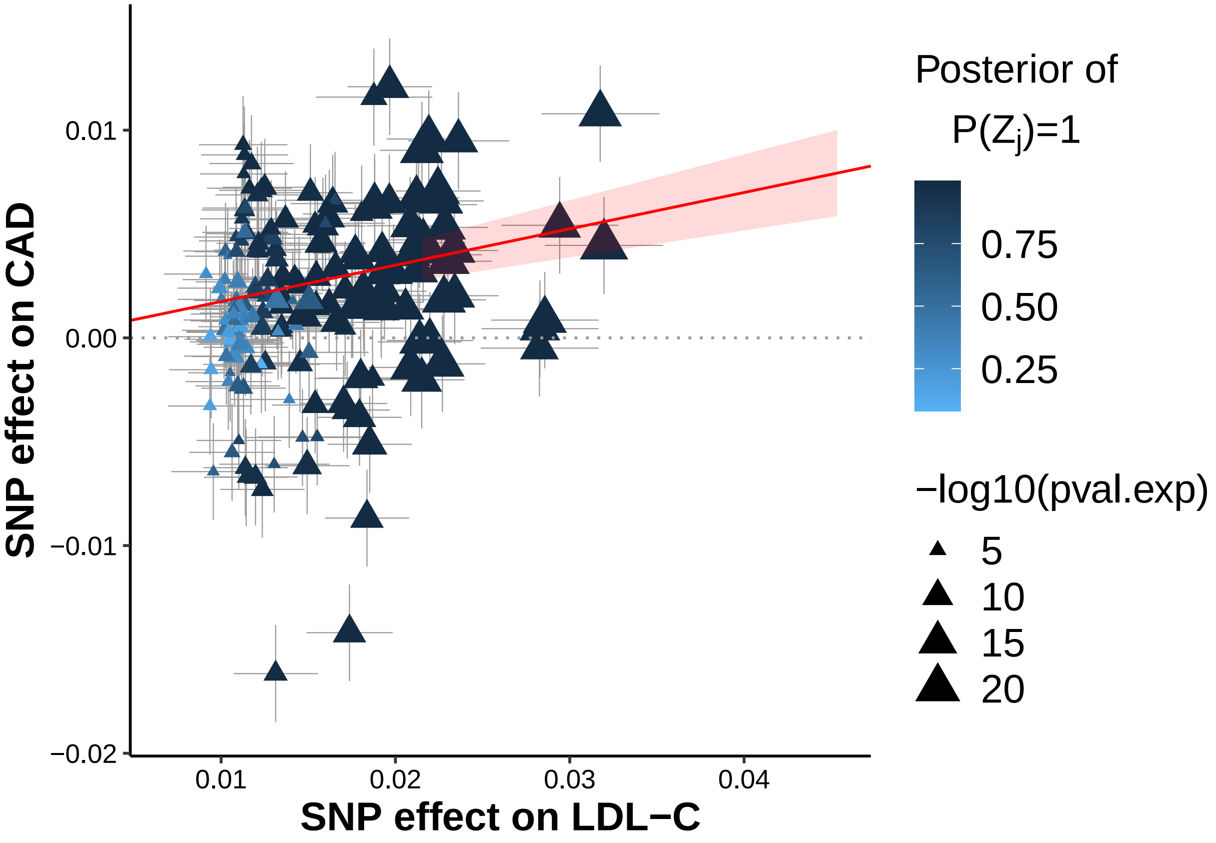
Step 5: Fit MRAPSS

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

```
MRres = MRAPSS(MRdat,
               exposure = "LDL-C",
               outcome = "CAD",
               Sigma_err = paras$Sigma_err,
               Omega = paras$Omega ,
               Threshold = Threshold)
#> ***********************************************************
#> MR test results of  LDL-C  on  CAD :
#> MR-APSS: beta =  0.175 beta.se =  0.0234 pval =  7.2627e-14 #SNPs=  177
#> Correlation parameter (rho) due to sample overlap :  0.01479736
#> Proportion of effective IVs with forground signals:  0.8050402
#> Variance component (Omega) for background model =
#>              [,1]         [,2]
#> [1,] 1.253563e-07 1.798647e-08
#> [2,] 1.798647e-08 5.433895e-08
#> Varaince component (Lambda) for foreground model =
#>              [,1]         [,2]
#> [1,] 6.907502e-06 0.000000e+00
#> [2,] 0.000000e+00 5.309594e-07
#> ***********************************************************
```

Visualize MRAPSS analysis results by “MRplot()”:

```
MRplot(MRres, exposure = "LDL-C", outcome = "CAD")
```



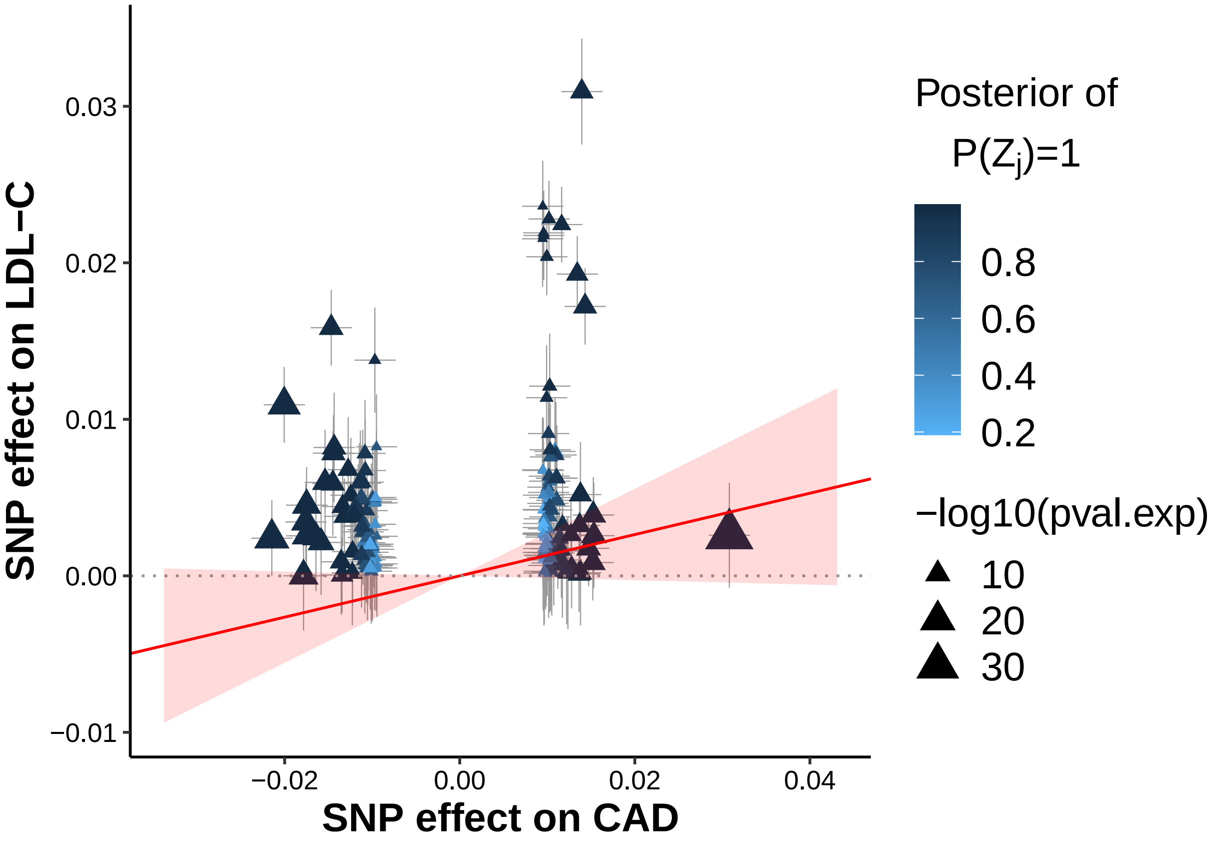
Try reverse direction MR analysis (CAD -> LDL-C)

Once you already have the estiamtes of parameters for LDL-C -> CAD and harmonised dataset from Step 3, you can also try reverse direction MR analysis (CAD -> LDL-C):

```r
# clump
MRdat_rev = clump(subset(paras$dat, pval.out < Threshold),
                  SNP_col = "SNP",
                  pval_col = "pval.out",
                  clump_kb = 1000,
                  clump_r2 = 0.001)
#> Clumping 1QNWim, 1338 variants
#> Removing 1187 of 1338 variants due to LD with other variants or absence from LD reference panel

colnames(MRdat_rev) = c("SNP", "A1","A2", "b.out", "b.exp", "se.out",
                        "se.exp", "pval.out", "pval.exp", "L2")
# run MRAPSS
MRres_rev = MRAPSS(subset(MRdat_rev, pval.exp < Threshold),
               exposure ="CAD",
               outcome = "LDL-C",
               Sigma_err = matrix(rev(paras$Sigma_err),2,2),
               Omega =  matrix(rev(paras$Omega),2,2) ,
               Threshold =  Threshold)
#> ***********************************************************
#> MR test results of  CAD  on  LDL-C :
#> MR-APSS: beta =  0.132 beta.se =  0.0745 pval =  7.6336e-02 #SNPs=  151
#> Correlation parameter (rho) due to sample overlap :  0.01479736
#> Proportion of effective IVs with forground signals:  0.7452034
#> Variance component (Omega) for background model =
#>              [,1]         [,2]
#> [1,] 5.433895e-08 1.798647e-08
#> [2,] 1.798647e-08 1.253563e-07
#> Varaince component (Lambda) for foreground model =
#>              [,1]        [,2]
#> [1,] 1.982906e-06 0.00000e+00
#> [2,] 0.000000e+00 2.87683e-06
#> ***********************************************************

MRplot(MRres_rev, exposure = "CAD", outcome = "LDL-C")
```


0.03
0.02
0.01
0.00
−0.01
SNP effect on LDL−C
−0.02
0.00
0.02
0.04
SNP effect on CAD
Posterior of P(Z_j)=1
0.8
0.6
0.4
0.2
−log10(pval.exp)
▲ 10
▲ 20
▲ 30