Robust Mendelian randomization method accounting for idiosyncratic and correlated pleiotropy with applications to stroke outcomes.

Qing Cheng

Introduction

This vignette provides an introduction to the *RMR.ICP* package. R package *RMR.ICP* implements RMR-ICP for Robust Mendelian randomization method accounting for idiosyncratic and correlated pleiotropy.

Install the development version of *RMR.ICP* by use of the 'devtools' package. Note that *RMR.ICP* depends on the 'Rcpp' and 'RcppArmadillo' package, which also requires appropriate setting of Rtools and Xcode for Windows and Mac OS/X, respectively. This package now depends on R (>= 3.5.0).

To install this package, run the following command in R

```
library(devtools)
install_github("QingCheng0218/RMR.ICP@main")
```

Fit RMR-ICP for independent SNPs using simulated data

In this section, we fit RMR-ICP using simulated data. We first generate genotype data using function *genRawGeno*:

```
rm(list = ls());
library(mvtnorm)
library(RMR.ICP)

h2a = 0.05; rho = 0; L = 100; M = 1; Alrate = 0.1;
dfA = 10; sigma2g = 0.01; delta = 1;

h2g = 0.1; b1 = 0.1; n1 = 50000; n2 = 50000;
maf = runif(M*L, 0.05, 0.5);
x = genRawGeno(maf, L, M, rho, n1 + n2);
x1 = x[1:n1,];
x2 = x[(n1+1):(n1+n2),];
x12 = x[1:(n1+n2),];
```

We use the following function *genSumStat* to generate the summary statistics.

```
# Generate the summary statistics.
SumStatres = genSumStat(x12, n1, n2, M, L, b1, Alrate, sigma2g, dfA, delta, h2a, h2g);
gammah_Indep = SumStatres$gammah;
se1_Indep = SumStatres$se1;
Gammah_Indep = SumStatres$Gammah;
se2_Indep = SumStatres$se2;
CHPindexTrue = SumStatres$CHPindex;

# Run RMR-ICP using independent SNPs.
```

```
result4indep = RMRICPindep(gammah_Indep, Gammah_Indep, se1_Indep, se2_Indep);
beta = result4indep$beta.hat;
se = result4indep$beta.se;
pvalue = result4indep$beta.p.value;

cat("The estimated effect of the exposure on outcome: ", beta);
```

```
## The estimated effect of the exposure on outcome:  0.09839605
```

```
cat("Standard error of beta1: ", se);
```

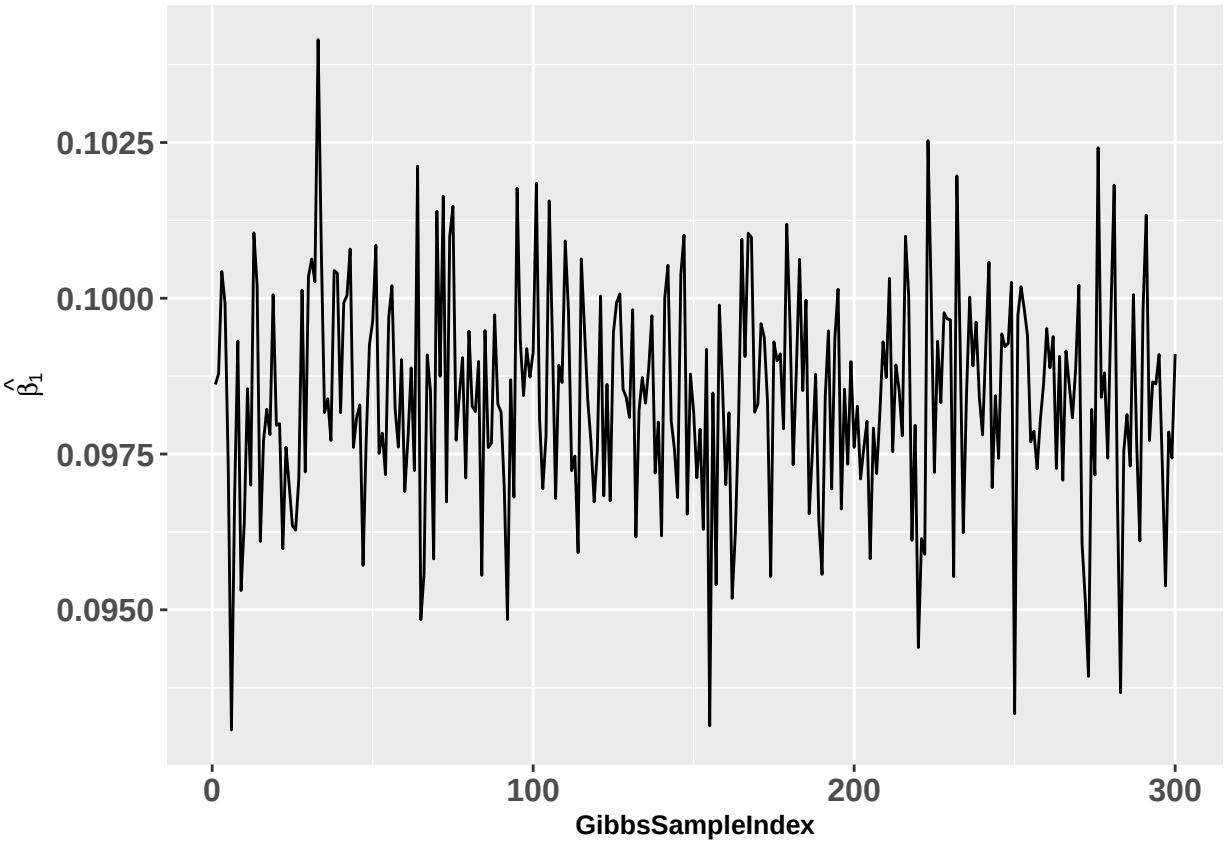
```
## Standard error of beta1:  0.001718228
```

```
cat("P-value for beta1: ", pvalue);
```

```
## P-value for beta1:  0
```

Check the convergence of Gibbs sampler using trace plot.

```
# Check the traceplot.
library(ggplot2);
traceplot(result4indep$Beta1res);
```



Adjust the hyper-prior parameters using the following codes.

```
opt = list(agm = 0, bgm = 0, aal = 0, bal = 0,
           a = 1, b = 1, v = 100, maxIter = 5000, thin = 10, burnin = 5000);
result4indep = RMRICPindep(gammah_Indep, Gammah_Indep, se1_Indep, se2_Indep, opt)
```

Specifically, $agm = 0$ and $bgm = 0$ correspond to σ_γ^2, while $aal = 0$ and $bal = 0$ correspond to σ_α^2. The

parameters a and b are the prior parameters for ω, and v is the prior parameter for W_k. *burnin* refers to the number of initial iterations discarded at the beginning of the Gibbs sampling, while *maxIter* denotes the number of iterations for Gibbs sampling after the burn-in phase. Finally, *thin* specifies the interval for recording the Gibbs sampling results.

Fit RMR-ICP for correlated SNPs using simulated data

```
rm(list = ls());
library(mvtnorm)
library(RMR.ICP)
h2a = 0.05; rho = 0.4; L = 200; M = 10; Alrate = 0.1;
dfA = 10; sigma2g = 0.01; delta = 1;
h2g = 0.1; b1 = 0.1;
maf = runif(M*L, 0.05, 0.5);

n1 = 50000; n2 = 50000; n3 = 4000; lambda = 0.85
x = genRawGeno(maf, L, M, rho, n1 + n2 + n3);
x1 = x[1:n1,];
x2 = x[(n1+1):(n1+n2),];
x12 = x[1:(n1+n2),];
x1 = x[1:n1,];
x2 = x[(n1+1):(n1+n2),];
x12 = x[1:(n1+n2), ];
x3 = x[(n1+n2+1):(n1+n2+n3), ];
block_inf = cbind(seq(1, M*L, M), seq(M, M*L, M));
block_inf1 = block_inf - 1;
R = Cal_block_SimR(block_inf1, x3, lambda)

SumStatres = genSumStat(x12, n1, n2, M, L, b1, Alrate, sigma2g, dfA, delta, h2a, h2g)

gammah_LD = SumStatres$gammah;
se1_LD = SumStatres$se1;
Gammah_LD = SumStatres$Gammah;
se2_LD = SumStatres$se2;
CHPindexTrue = SumStatres$CHPindex;

result4LD = try(RMRICPSim(gammah_LD, Gammah_LD, se1_LD, se2_LD, R, block_inf1));
beta = result4LD$beta.hat;
se = result4LD$beta.se;
pvalue = result4LD$beta.p.value;

cat("The estimated effect of the exposure on outcome: ", beta);
```

```
## The estimated effect of the exposure on outcome:  0.1007418
```

```
cat("Standard error of beta1: ", se);
```

```
## Standard error of beta1:  0.001738613
```

```
cat("P-value for beta1: ", pvalue);
```

```
## P-value for beta1:  0
```

Fit RMR-ICP for correlated SNPs using real dataset

We provide an example to illustrate the implementation of RMR-ICP for real data analysis. For Body Mass Index (BMI), we use summary statistics from chromosome 1, chromosome 2, and chromosome 3, as reported in Locke et al. (2015) [PMID: 25673413]. For Type 2 Diabetes (T2D), we use summary statistics from chromosomes 1, 2, and 3, as reported in Mahajan et al. (2018) [PMID: 30297969]. The following datasets should be prepared for analysis: `BMICHR1CHR2CHR3.txt`, `T2DCHR1CHR2CHR3.txt`, `UK10KCHR1LDhm3.RDS`, `UK10KCHR2LDhm3.RDS`, `UK10KCHR3LDhm3.RDS`, and `UK10Ksnpinforhm3.RDS`. These datasets can be downloaded from this link(https://drive.google.com/drive/folders/1oKuQ4fFf8kRJMT1WUlxniyHDnZEJPYYK?usp=drive_link). All coordinates are relative to the hg19 version of the human genome

```
rm(list = ls());
library(RMR.ICP);
library(ggplot2);
filepan <- vector("list", 22);
NumChr = 3;
for(i in 1:NumChr){
  filepan[[i]] <- paste0("UK10KCHR", i, "LDhm3.RDS");
}

fileexp = "BMICHR1CHR2CHR3.txt";
fileout = "T2DCHR1CHR2CHR3.txt";
snpinfo = "UK10Ksnpinforhm3.RDS";
```

`fileexp`, `fileout` are the data sets names for exposure and outcome, respectively. These two data sets must have the following format showd in Table 1, note that it must be tab delimited.

Table 1: Data format used for exposure and outcome data.

SNP	chr	BP	A1	A2	beta	se	pvalue
rs1000050	1	162736463	C	T	0.0002	0.0054	0.9705
rs1000073	1	157255396	A	G	0.0007	0.0038	0.8538
rs1000075	1	95166832	T	C	-0.0028	0.0040	0.4839
rs1000085	1	66857915	C	G	0.0020	0.0044	0.6494
rs1000127	1	63432716	C	T	-0.0019	0.0042	0.6510

`filepen` contians LD information about all of chromosome 22 variants, see Table 2. The column named r indicates the correlation between SNP1 and SNP2 estimated from the reference panel data: UK10K Project [Avon Longitudinal Study of Parents and Children (ALSPAC); TwinsUK] merged with 1000 Genome Project Phase 3 (3,757 samples with 989,932 SNPs). Here we just demonstrate use of *RMR.ICP* on three chromosomes, you can change `NumChr = 22` for the following analysis over all 22 autosomes.

Table 2: Data format used for reference panel.

CHR	BlockID	SNP1	SNP2	r
1	1	rs12562034	rs4040617	-0.12635628
1	1	rs12562034	rs2980300	-0.13086387
1	1	rs12562034	rs4475691	0.04401255
1	1	rs12562034	rs1806509	0.09783578
1	1	rs12562034	rs7537756	0.03716719

We also need an additional file named `snpinfo` , which is saved from reference panel data, to match the three data sets (exposure, outcome and panel data) and align effect sizes.

```
pva_cutoff = 5e-04;
lambad = 0.85;
data <- ReadSummaryStat(fileexp, fileout, filepan, snpinfo, pva_cutoff, lambad);

F4gammah <- data$ResF4gammah;
F4Gammah <- data$ResF4Gammah;
F4se1 <- data$ResF4se1;
F4se2 <- data$ResF4se2;
F4Rblock <- data$ResF4Rblock;
F4SNPs <- data$ResF4SNPchr;
L = length(F4Rblock);
cat("The 'RMR-ICP LD' Method start...","\n");
```

```
## The 'RMR-ICP LD' Method start...
```

```
RMRICPRes = RMRICP(F4gammah, F4Gammah, F4se1, F4se2, F4Rblock);
RMRICPbeta = RMRICPRes$beta.hat;
RMRICPse = RMRICPRes$beta.se;
RMRICPpvalue = RMRICPRes$beta.p.value;
cat("The estimated effect of BMI on T2D: ", RMRICPbeta);
```

```
## The estimated effect of BMI on T2D:  0.8320313
```

```
traceplot(RMRICPRes$Beta1res)
```