

# MR-LDP: a two-sample Mendelian randomization for GWAS summary statistics accounting linkage disequilibrium and horizontal pleiotropy

*Qing Cheng and Jin Liu*

*2019-07-15*

## Introduction

This vignette provides an introduction to the MR.LDP package. R package MR.LDP implements MR-LDP, a two-sample Mendelian randomization for GWAS summary statistics accounting linkage disequilibrium and horizontal pleiotropy. The package can be installed with the following commands:

```
library(devtools);  
install_github("QingCheng/MR.LDP");
```

Load the package using the following command:

```
library(MR.LDP);
```

## Fit MR-LDP using simulated data

We first generate genotype data using function *genRawGeno*:

```
library("mvtnorm");
```

```
## Warning: package 'mvtnorm' was built under R version 3.5.2
```

```
library("PDSCE");
```

```
## Warning: package 'PDSCE' was built under R version 3.5.2
```

```
set.seed(2019);  
genRawGeno <- function(maf, L, M, rho, n){  
  SIGMA = matrix(nrow=M,ncol=M)  
  for (i in 1:M){  
    for (j in 1:M){  
      SIGMA[i,j] = rho^(abs(i-j));  
    }  
  }  
  
  nsnp = L*M;  
  X = NULL;  
  for ( l in 1:L ){  
  
    index = (M*(l-1)+1):(M*l);  
    Aaprob = maf[index]^2.;  
    Aaprob = 2*maf[index]*(1-maf[index]);  
    quanti = matrix(c(1-Aaprob-Aaprob, 1- Aaprob),M,2);  
    Xt = rmvnorm(n, mean=rep(0,M), sigma=SIGMA, method="chol")  
    Xt2 = matrix(0,n,M);  
    for (j in 1:M){  
      cutoff = qnorm(quanti[j,]);  
      Xt2[Xt[,j] < cutoff[1],j] = 0;
```

```

        Xt2[Xt[,j] >= cutoff[1] & Xt[,j] < cutoff[2],j] = 1; ## attention
        Xt2[Xt[,j] >= cutoff[2],j] = 2;
    }
    X <- cbind(X,Xt2);
}
return(X)
}

rho = 0.4;
L = 1; M = 50; p = M*L; m = p; Alrate = 1;

n1 = 20000; n2 = 20000; n3 = 2500; lam = 0.055;
maf = runif(p, 0.05, 0.5);
G = genRawGeno(maf, L, M, rho, n1 + n2 + n3);

G1 = G[1:n1,];
G2 = G[(n1+1):(n1+n2),];
G12 = G[1:(n1+n2),];
G3 = G[(n1+n2+1):(n1+n2+n3),];

```

Estimate the covariance matrix using function *pdsoft*:

```

R0 = cor(G3);
R = pdsoft(R0, lam)$theta;
diag(R) = rep(1, p);
mask = kronecker(diag(L),matrix(1,M,M));
R = R*mask;

```

Generate the exposure data( $y$ ) and outcome data( $z$ ) with prespecified indirect( $h_y^2$ ) and direct( $h_z^2$ ) heritability based on

$$\mathbf{y} = \mathbf{G}_1\boldsymbol{\gamma} + \mathbf{U}_x\boldsymbol{\eta}_x + \mathbf{e}_1, \quad \mathbf{z} = \beta_0\mathbf{x} + \mathbf{G}_2\boldsymbol{\alpha} + \mathbf{U}_y\boldsymbol{\eta}_y + \mathbf{e}_2,$$

```

h2z <- 0.05; h2y <- 0.1; b0 <- 0.1; q <- 50;
u = matrix(rnorm((n1+n2) * q),ncol=q);

sigma2g <- 0.005;
gamma.nz = rnorm(m)*sqrt(sigma2g);
indx = sample(1:p,m);
gamma = numeric(p);
gamma[indx] = gamma.nz;

Su = matrix(c(1,0.8,0.8,1),nrow=2)
bu = rmvnorm(q,mean=rep(0,2), sigma = Su,method="chol")
by = bu[,1]; bz = bu[,2];
uby = u%*%by; ubz = u%*%bz;
uby = uby/sqrt(as.numeric(var(uby)/0.6));
ubz = ubz/sqrt(as.numeric(var(ubz)/0.2));

G12g = G12%*%gamma;

if(b0!=0){

```

```

h2ga = (h2y * ( 1 + b0^2))/(b0^2 * (1 - h2y));
gamma0 = gamma/sqrt(as.numeric(var(G12g)/h2ga));
G12g = G12%%gamma0;
}

yall = G12g + uby + rnorm(n1+n2)*as.numeric(sqrt(1-var(uby)));

# The direct effects on Z
h2yb = var(b0*yall);
h2al = (h2z + h2z*h2yb)/(1 - h2z)

sigma2a <- 0.005;
if(h2z==0){
  alpha0 = rep(0, m);
  G12a = G12%%alpha0;
}else{
  alno = floor(p*Alrate);
  alpha.nz <- rnorm(alno)*sqrt(sigma2a);
  # sparse setting for pleiotropy
  indxAL = sample(1:p,alno);
  alpha = numeric(p);
  alpha[indxAL] = alpha.nz;

  G12a = G12%%alpha;
  alpha0 = alpha/sqrt(as.numeric(var(G12a)/(h2al)));
  G12a = G12%%alpha0;
}

resz = ubz + rnorm(n1+n2)*as.numeric(sqrt(1-var(ubz)));
zall = b0*yall + G12a + resz;
H2a.res <- var(G12a)/var(zall);
H2g.res <- var(b0*G12g)/var(zall);

y = yall[1:n1];
z = zall[(n1+1):(n1+n2)];

```

We then conduct single-variant analysis to obtain the summary statistics.

```

gammahall = numeric(p); Gammahall = numeric(p);
sg2all = numeric(p); sG2all = numeric(p);
pval = numeric(p);
for (i in 1:p){
  fm = lm(y~1+G1[,i]);
  gammahall[i] = summary(fm)$coefficients[2,1];
  sg2all[i] = summary(fm)$coefficients[2,2]^2;
  pval[i] = summary(fm)$coefficients[2,4];

  fm = lm(z~1+G2[,i]);
  Gammahall[i] = summary(fm)$coefficients[2,1];
  sG2all[i] = summary(fm)$coefficients[2,2]^2;
}

index = 1:p
sig.indx = index

```

```
sg2 = sg2all[sig.indx];
sG2 = sG2all[sig.indx];
gammah = gammahall[sig.indx];
Gammah = Gammahall[sig.indx];
```

Initilize the parameters for MR-LDP algorithm.

```
epsStopLogLik <- 1e-7; maxIter <- 10000;
beta0 <- 0; sgga2 <- 1;
agm <- bgm <- aal <- bal <- 0.001;
gamma <- rep(0.01, p);
alpha <- rep(0.01, p);
sgga2 <- 1;
sgal2 <- 1;
model <- 2
```

Fit MR-LDP w/ and w/o constraint that  $\beta = 0$  as:

```
resHb = MRLDP_SimPXvb(gammah, Gammah, sqrt(sg2), sqrt(sG2),
                      gamma, alpha, beta0, sgga2, sgal2, R,
                      constr = 0, epsStopLogLik, maxIter, model);

resH0 = MRLDP_SimPXvb(gammah, Gammah, sqrt(sg2), sqrt(sG2),
                      gamma, alpha, beta0, sgga2, sgal2, R,
                      constr = 1, epsStopLogLik, maxIter, model);

tstat = 2*(resHb$tstat - resH0$tstat);
pval = pchisq(tstat, 1, lower.tail = F);
beta_hat = resHb$beta0;
```