R Notebook

If one SNP has a marginal effect on a phenotype, it is known as an SNP interaction displaying marginal effects. In some cases, however, each individual SNP has no effect on the phenotype, but the combination has a strong effect; this is known as SNP interactions displaying no marginal effects (INME)

Wei Pan https://www.biostat.umn.edu/~weip/prog/BasuPanGE11/simRareSNP.R

minor allele frequency (MAF)

```
########simulate SNPs from a latent multivariate Gaussian variable with
######## a CS(rho) correlation structure.

########Input:
#               ORs: association in OR between the causal SNPs and outcome;
#                   implicitly,  # of causal SNPs = length(ORs);
#               n0: # of controls;
#               n1: # of cases ;
#               nSNP: # of noise/marker SNPs to be generated;
#               rho: the parameter in CS(rho) for the latent multivariate
#                    Gaussian variable to be discretized to SNPs;
#                    rho=0 means all SNPs are independent;
#               MAF0slow, MAF0sup: MAFs for the causal SNPs from
#                                   Unif(MAF0slow, MAF0sup);
#               MAFslow, MAFsup: MAF's of the noise SNPs are drawn from
#                                 Unif(MAFslow, MAFsup);
#               p0: background disease prevalence; i.e. true logistic reg model's intercept=log(p0/(1-p
########Output: a list of the binary outcome Y (=0 or 1) and SNPs (=0, 1 or 2);
#               Y is a vector of length n=n1+n0; X is a matrix of n by nSNP;
#               and location indices for causal SNPs.

simRareSNP<-function(ORs, n0=500, n1=500, nSNP=0, rho=0,
                 MAF0slow=0.001, MAF0sup=0.01,
                 MAFslow=0.001, MAFsup=0.01, p0=0.05){

nSNP0<-length(ORs)
q<-nSNP+nSNP0

###############index for causal SNPs; mix causal and noise SNPs:
############### SNP j is causal iff SNP0index[j]=1 or -1; =0 o/w
signORs<-rep(1, nSNP0)
signORs[ORs<1]<-0-1
SNP0indx<-sample(c(signORs,  rep(0, nSNP)), q, replace = FALSE)

R<-matrix(1, nrow=q, ncol=q)
for(i in 1:q)
  for(j in 1:q)
    if (i!=j) R[i, j]<-rho
svd.R<-svd(R)
R1<-svd.R$u %*% diag(sqrt(svd.R$d))
```

```r
##background disease prev = p0
b0<-log(p0/(1-p0))
logORs<-log(ORs)

MAF0s<-runif(nSNP0, MAF0slow, MAF0sup)
MAFs1<-runif(nSNP, MAFslow, MAFsup)
cutoff0<-qnorm(MAF0s)
cutoff1<-qnorm(MAFs1)

cutoff<-rep(0, q)
cutoff[SNP0indx!=0]=cutoff0
cutoff[SNP0indx==0]=cutoff1

X<-matrix(0, nrow=n0+n1, ncol=q)
Y<-rep(0, n0+n1); Y[(n0+1):(n0+n1)]<-1
i<-1
#sampling controls:
while ( i <= n0){
  X0<-rnorm(q, 0, 1) #: X0 ~ MVN(0, I)
  X1<-R1 %*% X0    #: X1 ~ MVN(0, R)
  X2<-ifelse(X1<cutoff, 1, 0)
  X0<-rnorm(q, 0, 1) #: X0 ~ MVN(0, I)
  X1<-R1 %*% X0    #: X1 ~ MVN(0, R)
  X3<-ifelse(X1<cutoff, 1, 0)
  X4<-X2+ X3
  pr<-1/(1 + exp(-(b0 + sum(logORs * X4[SNP0indx!=0]))))
  Y1<-sample(c(0, 1), 1, prob=c(1-pr, pr))
  if (Y1==0){
    X[i, ]<-X4
    i<-i+1
    }
  }
#sampling cases:
while ( i <= n0+n1){
  X0<-rnorm(q, 0, 1) #: X0 ~ MVN(0, I)
  X1<-R1 %*% X0    #: X1 ~ MVN(0, R)
  X2<-ifelse(X1<cutoff, 1, 0)
  X0<-rnorm(q, 0, 1) #: X0 ~ MVN(0, I)
  X1<-R1 %*% X0    #: X1 ~ MVN(0, R)
  X3<-ifelse(X1<cutoff, 1, 0)
  X4<-X2+ X3
  pr<-1/(1 + exp(-(b0 + sum(logORs * X4[SNP0indx!=0]))))
  Y1<-sample(c(0, 1), 1, prob=c(1-pr, pr))
  if (Y1==1){
    X[i, ]<-X4
    i<-i+1
    }
  }

list(Y=Y, X=X, SNP0indx=SNP0indx)

}
```

```
 now <- Sys.time()
### but how to pick false pos correlated to a gene??
#--------------------------------------
## There is a coef for each gene plus the intercept
coef.pvalues.naive<-NULL
coef.pvalues.labeled<-NULL
num.iter<-5
ORs<-sample(c(.25,.5,2,4),100,replace = TRUE)
ORs<-rep(10,20)

for(i in 1:num.iter){


naive.rupture<-c(rep(0,300),rep(1,300))

foo<-simRareSNP(ORs=ORs
                , n0=285
                , n1=315
                , nSNP=200
                , rho=0,
                 MAF0slow=0.001, MAF0sup=0.05,
                 MAFslow=0.001, MAFsup=0.05, p0=0.05)

labeled.rupture<-foo$Y
predisposition<-foo$X

df<-data.frame(predisposition=predisposition
               ,naive.rupture=naive.rupture
               ,labeled.rupture=labeled.rupture)

m.naive<- glm(naive.rupture ~ predisposition, family="binomial",data=df)


m.labeled<- glm(labeled.rupture ~ predisposition, family="binomial",data=df)



coef.pvalues.naive<-c(coef.pvalues.naive,coef(summary(m.naive))[,4])
coef.pvalues.labeled<-c(coef.pvalues.labeled,coef(summary(m.labeled))[,4])


  } #end loop

pvalues.diff<-(coef.pvalues.labeled<0.05)==
                            (coef.pvalues.naive<0.05)

sum(pvalues.diff)/length(pvalues.diff)
```

```
## [1] 0.5833
```

```
difftime(Sys.time(),now)
```

```
## Time difference of 3.511 secs
```

```
Sys.sleep(2)
```

Time for this code chunk to run: 2.05122303962708