Multi-MRF User's Guide

Chen Lyu

2022-05-17

1. Introduction

The multi-trait methylation random field (multi-MRF) method is developed to detect methylation quantitative trait loci (mQTLs) by evaluating the joint association between a set of CpG sites and a set of genetic variants within a genomic region.

The proposed method has several advantages:

- It is a multi-trait method that allows flexible correlation structures between neighboring CpG sites (e.g. distance-based correlation).
- It is also a multi-locus method that integrates the effect of multiple common and rare genetic variants.
- It models the methylation traits with a beta distribution to characterize their bimodal and interval properties.

This document will include 1) an example pipeline for processing genetic and epigenetic data; 2) detailed description for multi-MRF function, input and output data with an example; and 3) an example of simulating methylation traits under varying scenarios.

1.1 Dependencies

The function has the following dependencies:

```
library(CompQuadForm)
library(betareg)
library(MASS)
library(glasso)
library(PearsonDS)
```

2. Data pre-processing

An example is presented to process genetic and epigenetic data for region-based association test. The users may choose appropriate data pre-processing strategies based on the need of their study.

2.1 Data description

The details of our application data can be found elsewhere [1]. In this example, the raw genotype data contains approximately 5 million SNPs using Illumina® Infinium HumanOmni5Exome BeadChip, and the methylation data contains approximately 450K CpG sites using Illumina HumanMethylation450 Beadchip.

[1] Li, M., et al. Mapping methylation quantitative trait loci in cardiac tissues nominates risk loci and biological pathways in congenital heart disease. BMC genomic data 2021;22(1):1-12.

2.2 Methylation data processing

For epigenomic data, we provided an exmaple using the Bioconductor package "minfi"[2] in R for data processing and quality control. Detailed instructions of using "minfi" can be found elsewhere: https://bioconductor.org/packages/devel/bioc/vignettes/minfi/inst/doc/minfi.html. Functional normalization was applied to raw intensities, which used internal control probes on each array to remove between-array technical variations. Beta values were produced to measure the methylation level of CpG sites, and intensities with detection p-values greater than 0.01 were set to missing. We further removed CpG sites with more than 5% missing values or with a SNP in the probe. Detailed instructions of using "minfi" can be found elsewhere.

The sample codes for methylation data processing:

```
library(minfi)

baseDir <- "./Data/Epigenetic/Baylor"
targets <- read.metharray.sheet(baseDir)
RGSet <- read.metharray.exp(targets = targets)
RGSet
manifest <- getManifest(RGSet)
manifest

MSet<-preprocessRaw(RGSet)
RSet <- ratioConvert(MSet, what = "both", keepCN = TRUE)
GRSet <- mapToGenome(RSet)
snps <- getSnpInfo(GRSet)
detP <- detectionP(RGSet);

QC<-minfiQC(MSet)
plotQC(QC$qc);
plotSex(QC$object);
pSex<-QC$qc[,'predictedSex',drop=F];
qcReport(RGSet)

manifest <- getManifest(RGSet)
manifest
head(getProbeInfo(manifest))

ratioSet <- preprocessFunnorm(RGSet)

gset <- mapToGenome(ratioSet)
gset <- addSnpInfo(gset)
gset <- dropLociWithSnps(gset)

beta <- getBeta(gset)
mval<-getM(gset)
annotation<-getAnnotation(gset)
detP<-detP[rownames(beta),colnames(beta)]
beta0.01<-beta;
beta0.01[detP>0.01]<-NA;
mval0.01<-mval;
mval0.01[detP>0.01]<-NA;
```

[2]Aryee, Martin J., et al. "Minfi: a flexible and comprehensive Bioconductor package for the analysis of Infinium DNA methylation microarrays." Bioinformatics 30.10 (2014): 1363-1369.

2.3 Genotype data processing

For genomic data, we provided an example of using PLINK 1.9 [3] for data processing. Detailed instructions of using PLINK 1.9 can be found elsewhere: https://www.cog-genomics.org/plink/. We conducted sex check of samples and removed samples with low call rates. We also removed variants with high missing rates or deviated from Hardy-Weinberg equilibrium among control samples (p-value < 0.0001). Detailed instructions of using plink can be found elsewhere.

The quality control process:

```
system('plink --bfile GWAS_Heart_20180930 --check-sex --noweb');
system('plink --bfile GWAS_Heart_20180930 --missing --noweb');

sex.genetic<-read.table('./plink.sexcheck',header=T,stringsAsFactors = F)
sex.epigenetic<-read.csv('./predictedSex_epigenetic.csv',header=T)
rownames(sex.genetic)<-sex.genetic[,1];
rownames(sex.epigenetic)<-sex.epigenetic[,2];
sex.genetic<-sex.genetic[order(sex.genetic[,1]),]
sex.epigenetic<-sex.epigenetic[order(sex.epigenetic[,2]),]
table(sex.genetic[,4],sex.epigenetic[,3])


imiss<-read.table('./plink.imiss',header=T,stringsAsFactors = F)
lmiss<-read.table('./plink.lmiss',header=T,stringsAsFactors = F)
hist(imiss$F_MISS,xlab='Missing Rate',main='Histogram by subjects');
hist(lmiss$F_MISS,xlab='Missing Rate',main='Histogram by SNPs');


system('plink --bfile GWAS_Heart_20180930 --mind 0.05 --geno 0.05 --hwe 0.0001
       --make-bed --out GWAS_Heart_miss_sub_0.05_miss_snp_0.05_20181001 --noweb');
sub.rm<-imiss[imiss$F_MISS>0.05,1]
sex.genetic<-sex.genetic[!sex.genetic$FID%in%sub.rm,]
sex.epigenetic<-sex.epigenetic[!sex.epigenetic$ID%in%sub.rm,]
table(sex.genetic[,4],sex.epigenetic[,3])
```

2.4 Extraction of genetic and epigenetic data based on gene regions

To conduct region-based association tests, gene units were defined based on the UCSC genome browser under the genome assembly of GRCh37/hg19. A candidate genomic region was defined as a gene unit along with its 7.5KB upstream and downstream sequences.

Extraction of CpG sites based on gene regions

- The sample input:
  - CpG file

```
##         Name   chr      pos       X1       X2
## 1 cg04185470 chr22 16123266 16048266 16198266
## 2 cg24521010 chr22 16155290 16080290 16230290
```

[3] Purcell, Shaun, et al. "PLINK: a tool set for whole-genome association and population-based linkage analyses." The American journal of human genetics 81.3 (2007): 559-575.

```
## 3 cg11055967 chr22 16155342 16080342 16230342
## 4 cg19807306 chr22 16155760 16080760 16230760
## 5 cg04110531 chr22 16156713 16081713 16231713
## 6 cg26186732 chr22 16156925 16081925 16231925
```

- Methylation value file

```
##         site      Sub1      Sub2      Sub3      Sub4      Sub5      Sub6
## 1 cg04185470 0.8812277 0.9012733 0.8963759 0.8779411 0.8990108 0.8633126
## 2 cg24521010 0.7956327 0.8251591 0.7618945 0.8492415 0.8374375 0.7678164
## 3 cg11055967 0.4799213 0.6537571 0.5245706 0.4526143 0.4587261 0.5032171
## 4 cg19807306 0.7564190 0.7704682 0.7814180 0.7776771 0.5658004 0.8054136
## 5 cg04110531 0.3670602 0.3398795 0.4100253 0.3945680 0.3865649 0.4103945
## 6 cg26186732 0.6540752 0.5344043 0.6572191 0.6660089 0.7086510 0.6097874
##        Sub7      Sub8      Sub9
## 1 0.8799290 0.9043663 0.9055452
## 2 0.7299243 0.8051142 0.8141544
## 3 0.4564843 0.5338213 0.5713929
## 4 0.7422387 0.8119899 0.8292434
## 5 0.3610477 0.4124629 0.4222119
## 6 0.6280941 0.6174620 0.7218438
```

- Information file based on annotation

```
##     V1     V2     V3 V4
## 1 chr1  11868  14409  +
## 2 chr1  14361  29370  -
## 3 chr1  30365  30503  +
## 4 chr1  34610  36081  -
## 5 chr1  69090  70008  +
## 6 chr1 134772 140566  -
##                                                                                                V5
## 1                                                         NR_046018&DDX11L1,NR_148357&LOC102725121
## 2 NR_024540&WASH7P,NR_106918&MIR6859-1,NR_107062&MIR6859-2,NR_107063&MIR6859-3,NR_128720&MIR6859-4
## 3               NR_036051&MIR1302-2,NR_036266&MIR1302-9,NR_036267&MIR1302-10,NR_036268&MIR1302-11
## 4                                                               NR_026818&FAM138A,NR_026820&FAM138F
## 5                                                                                NM_001005484&OR4F5
## 6                                                                               NR_039983&LOC729737
```

- Sample code for methylation data extraction (for chromosome 22):

```
# Extract the CpG sites (Beta value) that are located within the merged gene ranges
gene_ranges=read.table("refGene.merged.bed",header=F)
mythsite_selected=c()

i <- 22
mthinfo=read.table(paste("./Info_beta_CHR",i,".txt",sep=""),header = T)
gene_subranges=subset(gene_ranges,V1==paste("chr",i,sep=""))
dir.create(paste("./cpg_extract/methy_beta_extract/mthchr",i,sep = ""))
for(m in 1:nrow(gene_subranges)){
  for (j in 1:nrow(mthinfo)){
    if (mthinfo$pos[j]>=gene_subranges$V2[m] && mthinfo$pos[j]<=gene_subranges$V3[m]){
```

```
      mythsite_selected=rbind(mythsite_selected,mthinfo[j,])
    }
  }
  if(!is.null(mythsite_selected))
  {saveRDS(mythsite_selected,
           file=paste("./cpg_extract/methy_beta_extract/mthchr",i,"/",sprintf("%02d",i),
                      "_",sprintf("%09d",gene_subranges$V2[m]),".rds",sep = ""))}
  mythsite_selected=c()
}


#Extract Beta values of those CpG sites in the ranges, they can match
i <- 22
#read in files
ranges_in_chr=list.files(paste("./cpg_extract/methy_beta_extract/mthchr",i,"/",sep=""))
dir.create(paste("./values_extract/beta_extract/betachr",i,sep = ""))
beta=read.table(paste("./Beta_CHR",i,".txt",sep = ""),header=T)
for (j in ranges_in_chr){
  cpg_in_ranges=readRDS(paste("./cpg_extract/methy_beta_extract/mthchr",i,"/",j,sep = ""))
  #extract values for each range
  for (m in 1:nrow(cpg_in_ranges)){
    beta_extracted=c()
    beta_line=beta[which(beta[,1]==cpg_in_ranges[m,1]),]
    beta_extracted=rbind(beta_extracted,beta_line)
  }
  #save
  saveRDS(beta_extracted,paste("./values_extraxt/beta_extract/betachr",i,"/",
                               substr(j,1,12),".rds",sep = ""))
}
```

- The sample output:
  - CpG output

```
##         Name   chr      pos       X1       X2
## 2 cg24521010 chr22 16155290 16080290 16230290
## 3 cg11055967 chr22 16155342 16080342 16230342
## 4 cg19807306 chr22 16155760 16080760 16230760
## 5 cg04110531 chr22 16156713 16081713 16231713
## 6 cg26186732 chr22 16156925 16081925 16231925
## 7 cg19057994 chr22 16159072 16084072 16234072
```

- Methylation beta value output

```
##         site      Sub1      Sub2      Sub3      Sub4      Sub5      Sub6
## 2 cg24521010 0.7956327 0.8251591 0.7618945 0.8492415 0.8374375 0.7678164
## 3 cg11055967 0.4799213 0.6537571 0.5245706 0.4526143 0.4587261 0.5032171
## 4 cg19807306 0.7564190 0.7704682 0.7814180 0.7776771 0.5658004 0.8054136
## 5 cg04110531 0.3670602 0.3398795 0.4100253 0.3945680 0.3865649 0.4103945
## 6 cg26186732 0.6540752 0.5344043 0.6572191 0.6660089 0.7086510 0.6097874
## 7 cg19057994 0.7069089 0.7616192 0.7982753 0.7403681 0.7481393 0.6961960
##        Sub7      Sub8      Sub9
## 2 0.7299243 0.8051142 0.8141544
```

```
## 3 0.4564843 0.5338213 0.5713929
## 4 0.7422387 0.8119899 0.8292434
## 5 0.3610477 0.4124629 0.4222119
## 6 0.6280941 0.6174620 0.7218438
## 7 0.7109503 0.7273413 0.7709164
```

Extraction of genetic variants based on gene regions

- The sample input data:
  - GWAS plink files (i.e. bed, fam and bim);

```
##   FamID IndID PatID MatID sex phenotype
## 1  Sub1  Sub1     0     0   0         2
## 2  Sub2  Sub2     0     0   0         2
## 3  Sub3  Sub3     0     0   0         2
## 4  Sub4  Sub4     0     0   0         2
## 5  Sub5  Sub5     0     0   0         2
## 6  Sub6  Sub6     0     0   0         2
```

```
##   chr          ID GD position allele1 allele2
## 1  22 kgp24675479  0 16054713       A       G
## 2  22  kgp3171179  0 16055122       A       C
## 3  22  rs12157537  0 16114244       0       G
## 4  22 kgp24732566  0 16114555       0       C
## 5  22   rs2334336  0 16139442       0       G
## 6  22 kgp14987749  0 16152031       A       C
```

```
##      kgp24675479 kgp3171179 rs12157537 kgp24732566 rs2334336 kgp14987749
## Sub1           2          1          2           2         2           2
## Sub2           2          2          2           2         2           2
## Sub3           2          2          2           2         2           2
## Sub4           2          2          2           2         2           2
## Sub5           2          2          2           2         2           2
## Sub6           2          2          2           2         2           2
##      rs4819391 kgp15081860 rs4145526 kgp14998351
## Sub1         2           2         2           2
## Sub2         2           2         2           2
## Sub3         2           2         2           2
## Sub4         2           2         2           2
## Sub5         2           2         2           2
## Sub6         2           2         2           2
```

```
-   information file
```

```
##   V1       V2       V3 V4                                                  V5
## 1 22 16150528 16193009  -                                   NR_122113&DUXAP8
## 2 22 16157078 16172265  + NR_073459&BMS1P18,NR_073460&BMS1P17,NR_133911&BMS1P22
## 3 22 16199673 16231289  -                                 NR_132385&LINC01297
## 4 22 16256331 16287937  -                                  NM_001136213&POTEH
## 5 22 16274608 16277577  +                                 NR_046571&POTEH-AS1
## 6 22 16448823 16449804  -                                 NM_001005239&OR11H1
```

- The sample codes to extract the SNPs located within each genomic region from the genotype data on chromosome 22 (run on bash):

```
s=./GWAS
d=./gnr_snp_extract/snp_chr22
mkdir -p $d
while read c b e st g
do

 # output location
 out=${d}/$(printf "%02d" $c)_$(printf "%09d" $b)

 #from and to
 fr=$[b-7500]
 if [ $fr -lt 0 ]; then
  fr=0
 fi
 to=$[e+7500]

 #extraction
 plink --bfile $s --chr $c --from-bp $fr --to-bp $to --make-bed --out $out

 #maps
 if [ -e $out.bed ]; then
   echo -e "$c\t$b\t$e\t$g" > $out.txt
 fi

done < ./chr22.bed
```

- The sample output: plink files (i.e. bed, fam and bim)

```
##   FamID IndID PatID MatID sex phenotype
## 1  Sub1  Sub1     0     0   0         2
## 2  Sub2  Sub2     0     0   0         2
## 3  Sub3  Sub3     0     0   0         2
## 4  Sub4  Sub4     0     0   0         2
## 5  Sub5  Sub5     0     0   0         2
## 6  Sub6  Sub6     0     0   0         2
```

```
##   chr          ID GD position allele1 allele2
## 1  22 kgp24681384  0 51198868       G       A
## 2  22 kgp24711818  0 51198906       A       G
## 3  22 kgp24647791  0 51206832       0       G
## 4  22 kgp22805180  0 51208537       A       G
## 5  22 kgp24690301  0 51208562       A       G
## 6  22 kgp22802203  0 51211689       0       G
```

```
##      kgp24681384 kgp24711818 kgp24647791 kgp22805180 kgp24690301 kgp22802203
## Sub1           2           1           2           2           2           2
## Sub2           2           1           2           1           2           2
## Sub3           2           1           2           2           2           2
## Sub4           2           1           2           1           2           2
## Sub5           2           0           2           2           2           2
```

```
## Sub6           2           1           2           2           2           2
##      kgp24730272 kgp22823224 rs11912510 kgp24663393 kgp24715065 kgp22754451
## Sub1           1           1           2           2           2           2
## Sub2           1           2           2           2           2           2
## Sub3           2           1           2           2           2           2
## Sub4           1           2           2           2           2           2
## Sub5           0           1           2           2           1           1
## Sub6           0           2           2           2           2           2
##      kgp22743944
## Sub1           2
## Sub2           2
## Sub3           2
## Sub4           2
## Sub5           2
## Sub6           2
```

3. Multi-MRF function

Description

The multi-trait methylation random field (multi-MRF) method aims to detect methylation quantitative trait loci (mQTLs) by evaluating the joint association between a set of CpG sites and a set of genetic variants within a genomic region. The CpG sites and variants can be extracted based on gene regions using steps above.

Usage

- null.Multi.MRF (Y,distance,X=NULL,correlation,out_type)
- Multi.MRF (Z,result.null,similarity='GR', weights=1, resample=5000)

Arguments

Parameter	Explanation
Y	Vector of methylation trait in long format, e.g. beta value for CpG site 1-j for subject 1, followed by beta value for CpG site 1-j for subject 2, ..., and beta value for CpG site 1-j for subject i distance Data frame with sample ID and genomic location
X	Optional data frame for covariates, each column representing one covariate, need to be consistent with the long format of Y
correlation	The assumed correlation structure between CpG sites, choose from 'exchangeable' (i.e. exchangeable correlation) or 'distance-based' (i.e. distance-based correlation)
out_type	The assumed distribution of methylation trait, choose from 'normal' (i.e. normal distribution) or 'beta' (i.e. beta distribution)
Z	Data frame with each column representing one SNP, need to be consistent with the long format of Y. The genotype data takes numeric value of 0, 1 or 2 denoting the number of minor allele
results.null	The output from null.Multi.MRF()
similarity	The genotypic similarity between subjects. By default, genetic relationship (i.e. 'GR') [4] is used

Parameter	Explanation
weights	The optional weighting scheme to each SNP. For example, larger weights can be given to rare variants for greater effect size
resample	The optional number of re-sampling for model converge

Detail

The null model, i.e. null.Multi.MRF(), has to be run first to estimate the parameters under the null hypothesis when SNPs are not associated with the methylation traits. Then run the main function Multi.MRF() to fit the model considering the effect of genotype.

Value

The Multi.MRF function returns p value to determine significance level of multi-trait multi-locus association

3.1 Dataset

a. Trait data (n=100, nCpG=10), long format

- y0, y1 and y2 represent methylation traits (beta values), ranging from [0, 1] with each following a beta distribution
- y0 was simulated with no causal association with genotypes
- y1 was simulated to causally associated with 10% of SNPs in one direction (positive associations)
- y2 was simulated to causally associated with 10% of SNPs in two directions (50% positive associations and 50% negative associations)

```
trait <-read.table("Traits.txt", header=T)
dim(trait)
```

```
## [1] 1000    4
```

```
trait[1:10,]
```

```
##       ID         y0        y1        y2
## 1   sub1 0.09410764 0.3960321 0.7892338
## 2   sub1 0.09157423 0.5166719 0.7778248
## 3   sub1 0.13697977 0.4510917 0.8932701
## 4   sub1 0.11937421 0.5954122 0.7427562
## 5   sub1 0.17486752 0.3355338 0.8847770
## 6   sub1 0.09812228 0.4472801 0.7037891
## 7   sub1 0.08869038 0.4072040 0.8882596
## 8   sub1 0.08610057 0.5814680 0.8227450
## 9   sub1 0.10823488 0.4885453 0.7879667
## 10  sub1 0.13521490 0.4576444 0.8785440
```

b. Distance data (n=100, nCpG=10), long format

- distance column represents the BP information for each CpG site

```
distance <- read.table("Distance.txt", header=T)
dim(distance)
```

```
## [1] 1000    2
```

```
distance[1:10,]
```

```
##      ID distance
## 1  sub1  8117353
## 2  sub1  8124584
## 3  sub1  8117625
## 4  sub1  8124676
## 5  sub1  8119834
## 6  sub1  8123170
## 7  sub1  8119241
## 8  sub1  8119587
## 9  sub1  8120371
## 10 sub1  8121569
```

c. Genotype data (n=100, nsnp=56)

- Mixture of common and rare variants

```
geno <- read.table("Genotypes.txt", row.names = 1, header=T)
dim(geno)
```

```
## [1] 100  56
```

```
geno[1:6,1:10]
```

```
##      rs11869914 rs182310512 rs140099910 rs11651993 rs7222731 rs13339691
## sub1          1           0           0          1         1          0
## sub2          1           0           0          0         0          0
## sub3          0           0           0          1         0          0
## sub4          0           0           0          0         0          0
## sub5          0           0           0          0         0          0
## sub6          0           0           0          0         0          0
##      rs4792588 rs11078735 rs193014094 rs76592252
## sub1         0          1           0          0
## sub2         0          0           0          0
## sub3         0          1           0          0
## sub4         0          0           0          0
## sub5         0          0           0          0
## sub6         0          0           0          0
```

d. Covariate data (n=100, ncov=2)

- X1 is a quantitative covaiate, while X2 is a binary covariate

```
cov <- read.table("Covariates.txt", row.names = 1, header=T)
dim(cov)
```

```
## [1] 100   2
```

```
head(cov)
```

```
##              X1 X2
## sub1 -1.8598421  0
## sub2 -1.4112841  0
## sub3 -0.1813815  0
## sub4  0.9978638  0
## sub5  2.3312547  1
## sub6 -1.2871354  0
```

3.2 Example

```
library(CompQuadForm)
library(betareg)
library(MASS)
library(glasso)
library(PearsonDS)

source('Multi-MRF.R')

# Generate weights
# To detect rare variants of relatively large effect, higher weights are given to rare variants
MAF <- colMeans(geno)/2
wt<-dbeta(MAF,1,25)

# Align with the long format of traits data
GT <- apply(geno, 2, rep, each=10)
COV <- apply(cov, 2, rep, each=10)

# Assuming methylation traits follow a beta distribution and the correlation structure is
# distance-based (exchangeable correlation can also be applied)
obj.dist.b0<-null.Multi.MRF(Y=trait$y0, distance=distance,X=COV,
                            correlation='distance-based', out_type='beta');
obj.dist.b1<-null.Multi.MRF(Y=trait$y1, distance=distance,X=COV,
                            correlation='distance-based', out_type='beta');
obj.dist.b2<-null.Multi.MRF(Y=trait$y2, distance=distance,X=COV,
                            correlation='distance-based', out_type='beta');

p.dgrf.dist.b0 <-Multi.MRF(Z=GT, obj.dist.b0, weights=(wt^2))$pvalue
p.dgrf.dist.b0
```

```
##           [,1]
## [1,] 0.8191573
```

```
p.dgrf.dist.b1 <-Multi.MRF(Z=GT, obj.dist.b1, weights=(wt^2))$pvalue
p.dgrf.dist.b1
```

```
##            [,1]
## [1,] 0.00920559
```

```
p.dgrf.dist.b2 <-Multi.MRF(Z=GT, obj.dist.b2, weights=(wt^2))$pvalue
p.dgrf.dist.b2
```

```
##             [,1]
## [1,] 0.005933126
```

```
# Assuming methylation trait follows a normal distribution after logit transformation
# and the correlation is distance-based (exchangeable correlation can also be applied)
y.l0<-log(trait$y0/(1-trait$y0))
y.l1<-log(trait$y1/(1-trait$y1))
y.l2<-log(trait$y2/(1-trait$y2))

obj.dist.l0<-null.Multi.MRF(Y=y.l0, distance=distance,X=COV,
                            correlation='distance-based', out_type='normal');
obj.dist.l1<-null.Multi.MRF(Y=y.l1, distance=distance,X=COV,
                            correlation='distance-based', out_type='normal');
obj.dist.l2<-null.Multi.MRF(Y=y.l2, distance=distance,X=COV,
                            correlation='distance-based', out_type='normal');

p.dgrf.dist.l0 <-Multi.MRF(Z=GT, obj.dist.l0, weights=(wt^2))$pvalue
p.dgrf.dist.l0
```

```
##           [,1]
## [1,] 0.8628957
```

```
p.dgrf.dist.l1 <-Multi.MRF(Z=GT, obj.dist.l1, weights=(wt^2))$pvalue
p.dgrf.dist.l1
```

```
##             [,1]
## [1,] 0.009253149
```

```
p.dgrf.dist.l2 <-Multi.MRF(Z=GT, obj.dist.l2, weights=(wt^2))$pvalue
p.dgrf.dist.l2
```

```
##             [,1]
## [1,] 0.005849648
```

3.3 Multiple testing adjustment

To adjust for the multiple testing for the number of genomic regions, strategies can be applied such as Bonferroni correction or Benjamini–Hochberg's false discovery rate.

```
# Assuming a total of three genomics regions y0, y1, y2 are tested, multiple testing adjustment
# is needed to adjust the p values from multi-MRF
pval <- c(p.dgrf.dist.b0, p.dgrf.dist.b1, p.dgrf.dist.b2)

# Bonferroni correction
p.adjust(pval,method = 'bonferroni')
```

```
## [1] 1.00000000 0.02761677 0.01779938
```

```
# Benjamini-Hochberg's false discovery rate
p.adjust(pval,method = 'BH')
```

```
## [1] 0.81915731 0.01380838 0.01380838
```

4. Traits Simulation

- Input dataset: GT (genotype data); info (information for genotype)
- Input parameter: size (sample size); causal (proportion of causal mQTLs variants); correlation structures (i.e. exchangeable, autoregressive and distance-based); causal structures between mQTLs and methylation traits (i.e. "unique", "half-shared", and "all-shared")

```
library(MASS)

GT <- read.table("Genotypes.txt", header=T,sep='\t',stringsAsFactors=F)
info <- read.table("info.info", header=T,sep='\t',stringsAsFactors=F)
size<-100; causal<-0.1;causal_structure <- "unique"; correlation="exchangeable";
set.seed(20131018);

row_n <- sample(nrow(GT),size = size, replace = F)
GT_n <- GT[row_n,]
MAF_n <- colMeans(GT_n)/2
MAF_n <- MAF_n[MAF_n != 0]
snpname<-rownames(as.data.frame(MAF_n))

GT_n <- GT_n[,snpname]

# distance dataset
# Assuming 10 CpG sites for each ID, each CpG corrresponds to a distance
# To make sure the CpGs are within the same gene region with genotype, distance range
# from min(info$posi) to max(info$posi)
# each time, random choose 10kb as the gene region
# within 10kb, random selected 10 CpG sites
distance <- as.data.frame(matrix(rep(rownames(GT_n),each=10),byrow=T,ncol=1))
colnames(distance) <- "ID"
distance$distance <- 0
s <- floor(runif(1, min=min(info$posi), max=max((info$posi))))
e <-s+10000
cpg <- floor(runif(10, min=s, max=e))
for (j in 1:size){
  distance[(1+10*(j-1)):(10*j),2] <- cpg
```

```
}

# updated GT_n within the region
snpname2 <- info$SNP[info$posi >= s & (info$posi <= e)]
snpname2 <- snpname2[snpname2 %in% snpname]
GT_n <- GT_n[,snpname2]
MAF_n <- colMeans(GT_n)/2

# generate variance-covariance matrix based on correlation structure
if (correlation == "exchangeable"){vcov<-matrix(0.7,10,10); diag(vcov) <-1}

if (correlation == "autoregressive"){
  vcov<-matrix(1,10,10);
  for (k in 1:10){
    for (j in 1:10){
      di <- min(k,j)
      vcov[k,j] <- 0.7^(k-di)*0.7^(j-di)
    }}}

if (correlation == "exponential"){
  vcov<-matrix(1,10,10);
  dis <- distance[1:10,2]
  td <- 10000 # 10kb as region, can use average distance
  for (k in 1:10){
    for (j in 1:10){
      vcov[k,j] <- exp(-abs(dis[k]-dis[j])/td*3)
    }
  }
}

# generate multivariate error based on vcov matrix
error <- mvrnorm(n=size, rep(0,10), vcov)
error <- error/(10*max(error))

EFF<- -1/(MAF_n*(1-MAF_n))*0.01;
EFF1<-EFF;
EFF2<-EFF;

# wt: weights for snps
wt<-dbeta(MAF_n,1,25)
loc<-1:length(MAF_n);

nloc1<-floor(ncol(GT_n)*causal);
nloc2<-floor(nloc1*0.5);

# select causal snps for 10 traits
# varying causal structure
if (causal_structure == "unique"){
  sn1_10<-sample(loc,size=nloc1*10,replace=F);
  sn2_10<-sample(sn1_10,size=nloc2*10,replace=F);
}

if (causal_structure == "all shared"){
```

```
  sn1_10<-rep(sample(loc,size=nloc1,replace=F),10);
  sn2_10<-rep(sample(sn1_10,size=nloc2,replace=F),10);
}

if (causal_structure == "half shared"){
  sn1_5<-sample(loc,size=nloc1*6,replace=F);
  shared1<-sample(sn1_5,size=nloc1,replace=F);
  sn1_10<-c(rep(shared1,5),sn1_5[! sn1_5 %in% shared1]);
  sn2_5<-sample(sn1_10,size=nloc2*6,replace=F);
  shared2<-sample(1:length(sn2_5),size=nloc2,replace=F);
  sn2_10<-c(rep(shared2,5),sn2_5[-shared2]);
}

# generate methylation trait
# y0: no causal mQTLs
# y.l0: y0 in logit transform
# y1: in association with 10% causal mQTLs in one direction
# y.l1: y1 in logit transform
# y2: in association with 10% causal mQTLs in bi-direction
# y.l2: y2 in logit transform
y0 <- as.data.frame(matrix(rep(0,each=10*size),byrow=T,ncol=1));colnames(y0) <- "Y";
y1 <- y0;y2 <- y0;y.l0 <- y0; y.l1 <- y0; y.l2 <- y0;

for (j in 1:10){
  sn1<-sn1_10[(1+nloc1*(j-1)):(nloc1*j)];
  sn2<-sn2_10[(1+nloc2*(j-1)):(nloc2*j)];

  effect1<-EFF1;
  effect1[-sn1]<-0;
  effect2<-effect1;
  effect2[sn2]<-effect2[sn2]*(-1);

  N <- size

  eta0 <- log(0.1/(1-0.1))+error[,j];
  min0 <- min(eta0)
  max0 <- max(eta0)
  eta0 <- eta0*(2.19-2)/(max0-min0)+(-2*min0+2.19*max0)/(min0-max0)
  eta0[eta0 > 2.19] <- 2.19

  # eta1 for one direction
  # eta1 = beta1*x1+beta2*x2+...+betak*xk, make min = -2.19, max = c, c to control power
  c <- 1.5
  eta1<-rowSums(GT_n*matrix(effect1,nrow=nrow(GT_n),ncol=ncol(GT_n),byrow=T))+error[,j]
  min1 <- min(eta1)
  max1 <- max(eta1)
  eta1 <- eta1*(2.19+c)/(max1-min1)+(c*min1+2.19*max1)/(min1-max1)
  eta1[eta1 > 2.19] <- 2.19

  # eta2 for bi-direction
  # eta2 = beta1*x1+beta2*x2+...+betak*xk, make min = -2.19, max = d, d to control power
  d <- 2.5
  eta2<-rowSums(GT_n*matrix(effect2,nrow=nrow(GT_n),ncol=ncol(GT_n),byrow=T))+error[,j]
```

```
  min2 <- min(eta2)
  max2 <- max(eta2)
  eta2 <- eta2*(2.19+d)/(max2-min2)+(d*min2+2.19*max2)/(min2-max2)
  eta2[eta2 > 2.19] <- 2.19

  mu0<-exp(eta0)/(1+exp(eta0));
  mu1<-exp(eta1)/(1+exp(eta1));
  mu2<-exp(eta2)/(1+exp(eta2));

  phi <- 30

  a0 <- mu0*phi
  a1 <- mu1*phi
  a2 <- mu2*phi

  b0 <- (1-mu0)*phi
  b1 <- (1-mu1)*phi
  b2 <- (1-mu2)*phi

  tmp.y0<-rbeta(N,a0,b0);
  tmp.y1<-rbeta(N,a1,b1);
  tmp.y2<-rbeta(N,a2,b2);

  tmp.y.l0<-log(tmp.y0/(1-tmp.y0));
  tmp.y.l1<-log(tmp.y1/(1-tmp.y1));
  tmp.y.l2<-log(tmp.y2/(1-tmp.y2));

  y0[seq(j, nrow(y0), 10),1] <- tmp.y0;
  y.l0[seq(j, nrow(y0), 10),1] <- tmp.y.l0;
  y1[seq(j, nrow(y0), 10),1] <- tmp.y1;
  y.l1[seq(j, nrow(y0), 10),1] <- tmp.y.l1;
  y2[seq(j, nrow(y0), 10),1] <- tmp.y2;
  y.l2[seq(j, nrow(y0), 10),1] <- tmp.y.l2;

}

# covariates
# each individual has 10 repeated measures
x1<-rnorm(nrow(GT_n))
x2<-sample(c(0,1),size=nrow(GT_n),replace = T,prob=c(0.5,0.5));

# output
# methylation trait data
Y <- cbind(distance$ID, y0, y1,y2,y.l0,y.l1,y.l2)
colnames(Y) <- c("ID","y0","y1","y2","y.l0","y.l1","y.l2")
dim(Y)
```

```
## [1] 1000    7
```

```
Y[1:10,]
```

```
##       ID         y0        y1        y2      y.l0      y.l1      y.l2
## 1  sub98 0.05474468 0.7969733 0.9406861 -2.848775 1.3674837 2.7637656
```

```
## 2  sub98 0.07541321 0.7694122 0.8669260 -2.506365 1.2049951 1.8740481
## 3  sub98 0.12023739 0.7770645 0.9601137 -1.990184 1.2486407 3.1810200
## 4  sub98 0.08581672 0.7248798 0.9731240 -2.365817 0.9687978 3.5892772
## 5  sub98 0.05706912 0.7780584 0.8585213 -2.804730 1.2543872 1.8030623
## 6  sub98 0.26212519 0.8319895 0.8648901 -1.034952 1.5997937 1.8565138
## 7  sub98 0.07296076 0.6306739 0.6949352 -2.542074 0.5351090 0.8232942
## 8  sub98 0.04537130 0.7847764 0.7132767 -3.046443 1.2937216 0.9113517
## 9  sub98 0.11294957 0.8097426 0.9573457 -2.060960 1.4483387 3.1110368
## 10 sub98 0.09092765 0.5811398 0.8784098 -2.302361 0.3274543 1.9774569
```

```
# distance data
dim(distance)
```

```
## [1] 1000    2
```

```
distance[1:10,]
```

```
##       ID distance
## 1  sub98  8119405
## 2  sub98  8116668
## 3  sub98  8119972
## 4  sub98  8122675
## 5  sub98  8123347
## 6  sub98  8125145
## 7  sub98  8124836
## 8  sub98  8117200
## 9  sub98  8124959
## 10 sub98  8115564
```

```
# covariates data
X <- as.data.frame(cbind(x1,x2))
rownames(X) <- rownames(GT_n)
head(X)
```

```
##                  x1 x2
## sub98 -0.4408990715  0
## sub84 -0.0007421917  0
## sub86 -1.7681677072  1
## sub83  1.9540431648  1
## sub68 -0.8471850687  1
## sub18 -1.2346337419  0
```