methylMnM Tutorial

Yan Zhou, Bo Zhang, Nan Lin, BaoXue Zhang and Ting Wang

May 1, 2024

Contents

1 Introduction

M&M is developed for jointly analyzing MeDIP-seq and MRE-seq data, which are derived from methylated DNA immunoprecipitation (MeDIP) experiments followed by sequencing (MeDIP-seq) and methyl-sensitive restriction enzymes experiments for unmethylated CpGs (MRE-seq). We have implemented the M&M method via a set of R functions with the computational intensive parts written in C. We make a R package named methylMnM and give a tutorial for the package. The method consist three steps.

Step 1: Data Pre-processing;

- Calculate the CpG count of each window.
- Calculate the MRE CpG count of each window.
- Calculate MeDIP-seq tag count of each window of control and treatment samples.
- Calculate MRE-seq tag count of each window of control and treatment samples.

Step 2: Calculating p-values of each window by the M&M test;
Step 3: q-values for FDR control;
Step 4: Select Significants.

We use a real dataset to illustrate the usage of the methylMnM package. The programs can run under the Linux system and windows XP system. The R versions should larger than 2.12.0.

2 Preparations

Before installing methylMnM package, the user must install two other R packages, which can be done using the following commands.

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+     install.packages("BiocManager")
> BiocManager::install("edgeR")
> BiocManager::install("statmod")
> library(edgeR)
> library(statmod)
```

Next, to install the methylMnM package into your R environment, start R and enter:

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+     install.packages("BiocManager")
> BiocManager::install("methylMnM")
```

Then, the methylMnM package is ready to load.

```
> library(methylMnM)
```

3 Data format

In order to reproduce the presented methylMnM workflow, the package includes the example data sets `all_CpGsite_chr18.txt` (6M), `three_Mre_CpGsite_chr18.txt`(1.75M), `H1ESB1_MeDIP_18.extended.txt` (16.06M), `H1ESB2_Medip_18.extended.txt`(14.35M), `H1ESB1_MRE_18.extended.txt`(5.43M), and `H1ESB2_MRE_18.extended.txt` (4.98M) in the `extdata` subdirectory of the methylMnM package.

The files contain genomic regions from chromosome 18 only, as covered by short reads obtained from a MeDIP and MRE experiment of human H1ESB1 cells.

All input and output files are in a .bed format. The input file contain the following columns.

- the first coulumn is of type character and contains the chromosome of the region (e.g. chr1)
- the second column is of type numeric and contains the start position of the mapped read

- the third column is of type numeric and contains the stop position of the mapped read
- the fourth column is of type character and contains the strand information of the mapped read

For MRE-seq data, we need "+" and "-" strand information in mapping process, which is general located at the sixth column of the input file. Each row represents a mapped read. These information can be extracted from the output file(s) of common mapping tools. methylMnM counts chromosome sequence positions starting at 0. Some alignment tools output the mapped regions by interpreting the first base of a chromosome as 1.

4 Data Pre-processing

The methylMnM program requires users to provide the genome-wide MeDIP-seq and MRE-seq reads and the reference genome. The pre-processing step involves binning and the calculation of MeDIP-seq and MRE-seq count and CpG and MRE-specific count in every bin. The mainly steps are given as follows,

- Calculate the CpG count and MRE CpG count of each window.
- Calculate the MeDIP-seq and MRE-seq tag count of each window.

And we still need deal the data in the processing, such as blacklist, CNV and PCR.

- Remove windows which are overlap blacklist regions of genome-wide. The blacklist regions are not our concerned.
- Copy number variation (CNV) normalization. Some regions may be copy times than normal sample when the sample is cancer sample. Therefore, we should compare it at the same level.
- Polymerase Chain Reaction (PCR) cutoff for MRE-seq. We find that quantitative PCR may effect to test results. Therefore, we should remove the influence of PCR technology.

In this manual, we are using example data located in the extdata subdirectory of the methylMnM package. The path is

```
> datafile<-system.file("extdata",  package = "methylMnM")
> filepath<-datafile[1]
```

The CpG count, MRE-CpG count, MeDIP-seq count and MRE-seq count of each bin are stored at

```
> dirwrite<-paste(setwd(getwd()),"/",sep="")
```

4.1 CpG number of each bin

Compute the total CpG number of each bin with each CpG site. The CpG site should include at least three columns "chromosome", "start position" and "end position".

The output file is include four columns, that is "chromosome", "start position", "end position" and "CpG count". Also, the function output a report for some parameters.

```
> binlength<-500
> file.cpgsite<-paste(filepath,"/all_CpGsite_chr18.txt",sep="")
> writefile<-paste(dirwrite,"numallcpg.bed",sep="")
> reportfile<-paste(dirwrite,"report_numallcpg.txt",sep="")
> countcpgbin(file.cpgsite,file.blacklist=NULL,file.bin=NULL, writefile=writefile, reportf
```

The arguments of the function.

- `file.cpgsite`: The path of cpg site file.
- `file.blacklist`: The path of blacklist file (If we do not use the file, there will be defaulted as NULL).
- `file.bin`: The path of all cpg bin file. For computing the number of sequence tag of each window, we use the file as a normalization window position. (If we do not use the file, there will be defaulted as NULL).
- `writefile`:The path of output results. (If writefile=NULL, there will return the results back to main program.)
- `reportfile`: The path of output results of bin length, the number of bin, total reads before processing and total reads after processing.
- `binlength`: The length of each window.(Defaulted length is 500).

4.2 MRE-CpG number of each bin

Compute the MRE CpG number of each bin with MRE CpG sites. MRE CpG is some specific CpGs in genome-wide, such as "CCGG", "GCGC" and "CCGC". The specific CpG number is directly bound up with each experiment. The MRE CpG sites should include at least three columns "chromosome", "start position" and "end position". The output file is include four columns, that is "chromosome", "start position", "end position" and "MRE CpG count".

```
> file<-paste(filepath,"/three_Mre_CpGsite_chr18.txt",sep="")
> file1<-paste(filepath,"/all_CpGsite_chr18.txt",sep="")
> allcpgfile<-paste(dirwrite,"numallcpg.bed",sep="")
> five_Mre_CpGsite<-read.table(file, header=FALSE, as.is=TRUE)
> four_Mre_CpGsite<-five_Mre_CpGsite[five_Mre_CpGsite[,4]!="ACGT",]
> mrecpg.site<-four_Mre_CpGsite[four_Mre_CpGsite[,4]!="CGCG",]
> writefile<-paste(dirwrite,"three_mre_cpg.bed",sep="")
> countMREcpgbin(mrecpg.site,file.allcpgsite=file1,file.bin=allcpgfile,writefile=writefile
```

The arguments of the function.

- `mrecpg.site`: The data of mre-CpG site.
- `file.allcpgsite`: The path of all cpg site file.
- `file.bin`: The path of all bins file. For computing the number of sequence tag of each window, we use the file as a normalize window position. (If we do not use the file, there will be defaulted as NULL).
- `writefile`: The path of output result. (If writefile=NULL, there will return the results back to main program).
- `binlength`:The length of each window. (Defaulted length is 500).

4.3 Medip-seq and MRE-seq count of each bin

Transform H1ESB1 and H1ESB2 sample Medip-seq or MRE-seq tags to bin count. The Medip-seq or MRE-seq tags should include at least three columns "chromosome", "start position" and "end position". The output file is include four columns, that is "chromosome", "start position", "end position" and "count".

MeDIP-seq count of each bin of H1ESB1 is computed by follow code.

```
> file3<-paste(filepath,"/H1ESB1_MeDIP_18.extended.txt",sep="")
> allcpgfile<-paste(dirwrite,"numallcpg.bed",sep="")
> writefile<-paste(dirwrite,"H1ESB1_MeDIP_num500_chr18.bed",sep="")
> reportfile<-paste(dirwrite,"H1ESB1_MeDIP_num500_chr18_report.txt",sep="")
> countMeDIPbin(file.Medipsite=file3,file.blacklist=NULL,file.bin=allcpgfile,file.CNV=NULL
```

- `file.Medipsite`: The path of sequence tags file.
- `file.blacklist`: The path of blacklist file (If we do not use the file, there will be defaulted as NULL).
- `file.bin`: The path of all cpg bin file. For computing the number of sequence tag of each window, we use the file as a normalization window position. (If we do not use the file, there will be defaulted as NULL).
- `file.CNV`: If need, we should input CNV file to normalize count of each bin.
- `writefile`:The path of output results. (If writefile=NULL, there will return the results back to main program.)
- `reportfile`: The path of output results of bin length, the number of bin, total reads before processing and total reads after processing.
- `binlength`: The length of each window.(Defaulted length is 500).

MRE-seq count of each bin of H1ESB1 is computed by follow code.

```
> file4<-paste(filepath,"/H1ESB1_MRE_18.extended.txt",sep="")
> writefile<-paste(dirwrite,"H1ESB1_MRE_num500_chr18.bed",sep="")
> reportfile<-paste(dirwrite,"H1ESB1_MRE_num500_chr18_report.bed",sep="")
> countMREbin(file.MREsite=file4,file.blacklist=NULL, file.bin=allcpgfile,file.CNV=NULL, c
```

- `file.MREsite`: The path of sequence tags file.
- `file.blacklist`: The path of blacklist file (If we do not use the file, there will be defaulted as NULL).
- `file.bin`: The path of all cpg bin file. For computing the number of sequence tag of each window, we use the file as a normalization window position. (If we do not use the file, there will be defaulted as NULL).
- `file.CNV`: If need, we should input CNV file to normalize count of each bin.
- `cutoff`: The critical value of PCR. (If we do not use the critical value, there will be defaulted as 0.)
- `writefile`:The path of output results. (If writefile=NULL, there will return the results back to main program.)
- `reportfile`: The path of output results of bin length, the number of bin, total reads before processing and total reads after processing.
- `binlength`: The length of each window.(Defaulted length is 500).

MeDIP-seq count of each bin of H1ESB2 is computed by follow code.

```
> file5<-paste(filepath,"/H1ESB2_Medip_18.extended.txt",sep="")
> allcpgfile<-paste(dirwrite,"numallcpg.bed",sep="")
> writefile<-paste(dirwrite,"H1ESB2_MeDIP_num500_chr18.bed",sep="")
> reportfile<-paste(dirwrite,"H1ESB2_MeDIP_num500_chr18_report.txt",sep="")
> countMeDIPbin(file.Medipsite=file5,file.blacklist=NULL,file.bin=allcpgfile,file.CNV=NULL
```

MRE-seq count of each bin of H1ESB2 is computed by follow code.

```
> file6<-paste(filepath,"/H1ESB2_MRE_18.extended.txt",sep="")
> writefile<-paste(dirwrite,"H1ESB2_MRE_num500_chr18.bed",sep="")
> reportfile<-paste(dirwrite,"H1ESB2_MRE_num500_chr18_report.txt",sep="")
> countMREbin(file.MREsite=file6,file.blacklist=NULL, file.bin=allcpgfile,file.CNV=NULL, c
```

5 p-values of M&M test

In order to detect different methylated bins, we should calculate p-value of each bin. The below codes are calculate p-value of each bin with M&M method. The input files which include "datafile", "cpgfile" and "mrecpgfile" are should have been generated by Step 1. The output file "writefile" will own eleven columns, that is, "chr", "$chrSt$", "$chrEnd$", "$Medip1$", "$Medip2$", "$MRE1$","$MRE2$", "cg", "$mrecg$" "$pvalue$" and "t". We also output a report file which will include parameters "$s1/s2$", "$s3/s4$", "$N1$", "$N2$", "$N3$", "$N4$", "$c1$", "$c2$", "$Number\ of\ windows$" and "$Spend\ time$".

```
> datafile1<-paste(dirwrite,"H1ESB1_MeDIP_num500_chr18.bed",sep="")
> datafile2<-paste(dirwrite,"H1ESB2_MeDIP_num500_chr18.bed",sep="")
> datafile3<-paste(dirwrite,"H1ESB1_MRE_num500_chr18.bed",sep="")
> datafile4<-paste(dirwrite,"H1ESB2_MRE_num500_chr18.bed",sep="")
```

```
> datafile<-c(datafile1,datafile2,datafile3,datafile4)
> chrstring<-NULL
> cpgfile<-paste(dirwrite,"numallcpg.bed",sep="")
> mrecpgfile<-paste(dirwrite,"three_mre_cpg.bed",sep="")
> writefile<-paste(dirwrite,"pvalH1ESB1_H1ESB2_chr18.bed",sep="")
> reportfile<-paste(dirwrite,"report_pvalH1ESB1_H1ESB2_chr18.txt",sep="")
> MnM.test(file.dataset=datafile,chrstring=chrstring,file.cpgbin=cpgfile,file.mrecpgbin=mr
```

The arguments of the function.

- `file.dataset`: The files path of sample. (datafile should be c(datafile1,datafile2,datafile3,datafile4), where datafile1 and datafile2 are path of Medip-seq data, datafile3 and datafile4 are path of MRE-seq data).
- `chrstring`: The chromosome should be test. If "chrstring=NULL", we will test all chromosome.
- `file.cpgbin`: The file path of all CpG number of each bin.
- `file.mrecpgbin`: The file path of MRE-CpG number of each window (If NULL, mrecpgfile will equal to cpgfile).
- `writefile`:The file path of output result. (If writefile=NULL, there will return the results back to main program).
- `reportfile`: The path of output results of bin length, the number of bin, total reads before processing and total reads after processing.
- `mreratio`:The ratio of total unmethylation level with total methylation level (Defaulted mreratio is $3/7$).
- `method`: Option different data for the test. If we using $method = "XXYY"$, that means we calculate p-value with $P(\mid c_1X_{1i}Y_{2i} - c_2X_{2i}Y_{1i} \mid > t_i | X_{1i} + X_{2i} + Y_{1i} + Y_{2i} = n_i)$. Else is $P(\mid c_1X_{1i}Z_{2i} - c_2X_{2i}Z_{1i} \mid > t_i | X_{1i} + X_{2i} + Z_{1i} + Z_{2i} = n_i)$. `psd and mkadded`: The parameters of pseudo count, which *psd* added to Medip-seq and MRE-seq count and *mkadded* added to all CpG and MRE CpG (We set psd=2 and mkadded=1 as defaulted for robust)
- `a and top`: Cut-off for recalculating p-value with multinomial distribution when normal p-values smaller than a and the sum of observations smaller than top.
- `cut`: Cut-off for recalculating p-value with multinomial distribution when the sum of observations smaller than cut.

6 q-values

In order to control FDR, we estimate q-values with p-values. The input file is the file which have been generated by Step 2. And the output file is just add a q-value column to the input file. The function is,

```
> datafile<-paste(dirwrite,"pvalH1ESB1_H1ESB2_chr18.bed",sep="")
> writefile<-paste(dirwrite,"q_H1ESB1_H1ESB2_chr18.bed",sep="")
> reportfile<-paste(dirwrite,"report_q_H1ESB1_H1ESB2_chr18.bed",sep="")
> MnM.qvalue(datafile,writefile,reportfile)
```

The arguments of the function.

- `datafile`: The path of p-values file.
- `writefile`: The file path of output result. (If writefile=NULL, there will return the results back to main program).
- `reportfile`: The path of output results of bin length, the number of bin, total reads before processing and total reads after processing.

7 Select Significants

After getting q-values, we choose DMRs with p-values or q-values. The input file is the file which have been generated by Step 3. And the output file is some rows of the input file which meet the cutoff. The function is,

```
> file<-paste(dirwrite,"q_H1ESB1_H1ESB2_chr18.bed",sep="")
> frames<-read.table(file, header=TRUE,sep="\t", as.is=TRUE)
> DMR<-MnM.selectDMR(frames =frames , up = 1.45, down =1/1.45, p.value.MM = 0.01, p.value.
> writefile<-paste(dirwrite,"DMR_e5_H1ESB1_H1ESB2_chr18.bed",sep="")
> write.table(DMR, writefile,sep="\t", quote=FALSE,row.names=FALSE)
```

The arguments of the function.

- `frames`: The p-value or q-value data.
- `up`: The ratio of Medip1/Medip2 should be larger than "up" value if we call it significant.
- `down`: The ratio of Medip1/Medip2 should be smaller than "down" value if we call it significant.
- `p.value.MM`: The p-value of the bin which use MM test should be smaller than "p.value.MM" if we call it significant.
- `p.value.SAGE`: The p-value of the bin which use SAGE test should be smaller than "p.value.SAGE" if we call it significant.
- `q.value`: The q-value of the bin should be smaller than "q.value" if we call it significant.
- `cutoff`: We should use p-value or q-value cutoff to detect DMRs (If cutoff="q-value", then we use q-value to detect DMRs, else we use p-value).
- `quant`: The rank of absolute value of (Medip1-Medip2) should be larger than "quant" value if we call it significant.

8 Concluding Remarks

In our opinion, The package (methylMnM) provides several helpful functionalities for analyzing MeDIP-seq and MRE-seq data in reasonable time compared to other available approaches. Nevertheless, there are some limitations that have to be addressed in the future. Main issues are

- Because methylMnM (M&M) processes full genome data at once, methylMnM (M&M) needs a lot of memory at 3.1 section. But calculate p-value and q-value just spend 2-3GB memory for full genome.
- We can control run time by adjusting *top* parameter in function $MMtest()$. Larger *top* value will take more time.
- The MM.selectSignificants() function is a novel approach for the identifcation of DMRs. We can use p-value or q-value, which accompanied some other standards (`up`, `down` and `quant`), to select DMR.