hmmDMR User Guide

Tieming Ji, jit@missouri.edu
May, 2018

1. Introduction

hmmDMR is an R package for identifying differentially methylated regions (DMRs) between case and control groups using whole genome bisulfite sequencing (WGBS) or reduced representative bisulfite sequencing (RRBS) experiment data. In this user guide, we will show step-by-step how to use the hmmDMR package to find DMRs.

Before we start to use the package, it is helpful to understand that the hmmDMR package uses a Bayesian hidden Markov model (HMM) for detecting DMRs. It fits a Bayesian HMM for each chromosome. The final output of hmmDMR are DMRs with start and end position in a given chromosome, directions of the DMRs (hyper- or hypo-), and the numbers of CpGs in the DMRs. The R package contains the following four functions: (1) read.process() function is to read in data; (2) initial.value() function set the initial values for the Expectation-Maximization (EM) algorithm to estimate parameters in the Bayesian HMM; (3) EM() function execute the estimation procedure for the Bayesian HMM and infer the best sequence of methylation states; (4) PostAdjustment() function allows researchers to put extra requirements of DMRs such as the minimum length of a DMR, the minimum number of CpGs in a DMR, and the maximum distance (in base pairs) between any two adjacent CpGs. In the next a few sections, we will explain the usage of these four functions one by one.

The statistical method featuring this algorithm is currently under review. Please cite the paper as follows.

Ji, Tieming (2018) A Bayesian Hidden Markov Model for Detecting Differentially Methylated Regions. Under Review.

2. Read in the methylation data

The analysis starts by reading in the methylation data from either WGBS or RRBS experiments. For example, suppose we have n1 replicates from the control group and n2 replicates from the case group. We do not require replication in either the control or case group; i.e., n1>=1 and n2>=1.

The "read.process()" function reads in data and transforms observations into data that the Bayesian HMM can directly use. It has six parameters: pos, norm.m, norm.um, abnorm.m, abnorm.um, and bin.size.
(1) pos: A vector containing CpG positions;
(2) norm.m: A matrix contains methylated read count data of control (or normal) group. Each column of the matrix represents a replicate and each row represents a CpG positon;
(3) norm.um: A matrix contains unmethylated read count data of normal group;
(4) abnorm.m: A matrix contains methylated read count data of abnormal group;
(5) abnorm.um: A matrix contains unmethylated read count data of abnormal group;

(6) bin.size: An integer for bin size. Default to 40.
Using Chen et al. (2015, 2017)'s study on large offspring syndrome (LOS) as an example. There are four replicates in the case (LOS) group and four replicates in the control group. The raw FASTQ files of the WGBS experiment from this study are available at Gene Expression Omnibus with accession no. GSE93775. We use chromosome 29 of this study as an example.

```
(1) pos=c(271, 331, 363, 386, 418, 464, ...)
(2) norm.m:
     [,1] [,2] [,3] [,4]
[1,]    8    7   12   10
[2,]    4    4    2    4
[3,]    0    1    0    4
[4,]    2    2    0    2
[5,]    1    1    1    1
[6,]    8    0    0    7
...
(3) norm.um:
     [,1] [,2] [,3] [,4]
[1,]    4    7    2    4
[2,]   12   11   11   10
[3,]   10   10    8    7
[4,]    8   11   10   13
[5,]    7   11    6   17
[6,]    8    9    7    8
...
(4) abnorm.m:
     [,1] [,2] [,3] [,4]
[1,]   10    7   10   13
[2,]    6    2    6    8
[3,]    3    0    3    0
[4,]    0    1    1    0
[5,]    1    1    2    2
[6,]    6    4    8    7
...
(5) abnorm.um:
     [,1] [,2] [,3] [,4]
[1,]    6    3    3    3
[2,]    9    5    6   12
[3,]   12   11    8   20
[4,]    8   13   12   15
[5,]   10   12   12   19
[6,]    8    7    6   14
...
(6) bin.size=40
```

```
obs <-  read.process(pos, norm.m, norm.um, abnorm.m, abnorm.um, bin.size)
```

"obs" is a matrix shown as follows. Column "o" is the methylation rate difference between abnormal and normal groups after logistic transformation at each CpG site.

$$o = \log(abnorm.p/(1-abnorm.p)) - \log(norm.p/(1-norm.p)).$$

Column "dist" shows the distance between the start of a bin and the start of a bin ahead of it. For the first bin, "dist" shows the position of the first bin in the chromosome. Column "abnorm.p" shows the average of methylation rate across replicates in the abnormal group at each CpG site. Column "norm.p" shows the average methylation rate across replicates in the normal group at each CpG site. Column "start" is the start position of a bin, and column "end" is the end position of a bin.

```
          o   dist   abnorm.p    norm.p start end
1  0.1853336  280  0.7118644 0.6724138   241 280
2  0.7077460   80  0.4137931 0.2580645   321 360
3 -0.4634234   40  0.1043478 0.1562500   361 400
4  0.0415490   40  0.1269841 0.1224490   401 440
5  0.3780661   40  0.4218750 0.3333333   441 480
6  0.5877867   80  0.1730769 0.1041667   521 560
```

Our Bayesian HMM models on the observations in the column "o". We use EM algorithm to find the best sequence of hidden states by maximizing the expected likelihood given observations in the column "o".

3. Set initial values for Bayes HMM

Function initial.value() takes the output from read.process() function to set the initial values of parameters for the Bayes HMM model fitting. This step does not need any user interaction. The algorithm automatically set a good set of initial parameter values.

```
initial.para <- initial.value(obs)
```

"initial.para" contains the following parameters: p0, p1, and p2 are the probability distribution for the first bin of the Markov chain. mu.pos and mu.neg are the mean methylation rates of hyper- and hypo-bins, respectively. sd0, tao1, and tao2 are the standard deviation for the normal, hyper-, and hypo-bins, respectively. At last, tran.p contains a vector of parameters in the transition matrix.

4. EM algorithm to estimate Bayes HMM parameters

Function EM() takes the output from read.process() and initial.value() functions, and executes the EM algorithm to estimate model parameters. This step does not require any user interaction. As long as the input data are given, the algorithm automatically executes and finds the best sequencing of hidden methylation states.

```
em.o <- EM(initial.para, obs)
```

The output "em.o" is a list contains two parts, "res" and "para". "res" is a matrix that contains predicted methylation states; "para" is a list that contains model parameters after EM algorithm converges.

"res" is the same with "obs" except that it has an additional column at the end of the matrix "direction" showing the predicted methylation states by our Bayesian HMM. "0" indicates normal bin; "1" indicates hyper-bin; "2" indicates hypo-bin.

```
          o   dist   abnorm.p    norm.p start end   direction
1  0.1853336  280  0.7118644 0.6724138   241 280           1
2  0.7077460   80  0.4137931 0.2580645   321 360           1
3 -0.4634234   40  0.1043478 0.1562500   361 400           0
```

```
4  0.0415490   40  0.1269841 0.1224490   401 440         0
5  0.3780661   40  0.4218750 0.3333333   441 480         0
6  0.5877867   80  0.1730769 0.1041667   521 560         0
```

5. Find and report DMRs

Post adjustment is often necessary for methylation data analysis since biological data is often of high noise and high variation. Besides, biological researchers are often interested in DMRs with certain requirements. Function PostAdjustment() takes the output from EM() function and refines DMR results.

```
dmr <- PostAdjustment(em.o, pos, min.length=1000, min.CpGs=10, max.gap=300)
```

"em.o" is the output from EM() function. pos is a vector contains CpG positions, which is the same with the input pos for the read.process() function. min.length is the minimum length required for a DMR. min.CpGs is the minimum number of CpGs contained in an identified DMR. max.gap is the maximum gap in base pairs between any adjacent two CpGs within one DMR.

To run this function we need users to input min.length, min.CpGs, and max.gap.

The output of PostAdjustment(), i.e., "dmr", is a list that contains two parts: "dmr.res" and "region". "dmr.res" is the same with the "res" from the EM() output except that the last column "direction" is refined to meet the requirements of "min.length", "min.CpGs", and "max.gap". The second part "region" in the output contains the final DMRs with the start and end position of a DMR, the length of the DMR, and the number of CpGs in it.

```
dmr$region

  region.cnt region.start region.end region.state num.CpGs length
1          1      2069041    2070280         hypo       15   1239
2          2     12343081   12344320        hyper       17   1239
3          3     25002401   25003440         hypo       66   1039
4          4     35100321   35101360        hyper       15   1039
5          5     36306041   36307520         hypo       27   1479
6          6     36979801   36980840         hypo       14   1039
7          7     42175641   42176800         hypo       41   1159
8          8     48698441   48699560        hyper       27   1119
9          9     49553441   49555240         hypo      147   1799
```

References

Chen, Z., Hagen, D. E., Elsik, C. G., Ji, T., Morris, C. J., Moon, L. E., and Rivera, R. M. (2015) Characterization of global loss of imprinting in fetal overgrowth syndrome induced by assisted reproduction. Proceedings of the National Academy of Sciences 112, 4618-4623.

Chen, Z., Hagen, D. E., Ji, T., Elsik, C. G., and Rivera, R. M. (2017) Global misregulation of genes largely uncoupled to DNA methylome epimutations characterizes a congenital overgrowth syndrome. Scientific Reports 7, 12667.