The hummingbird Package

Eleni Adam

5/14/2020

Introduction

hummingbird is a 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.

The hummingbird package uses a Bayesian hidden Markov model (HMM) for detecting DMRs. It fits a Bayesian HMM for each chromosome. The final output of hummingbird is the DMRs with start and end position in a given chromosome, directions of the DMRs (hyper- or hypo-), and the number of CpGs in the DMRs.

Functions

The hummingbird package contains the following three functions:

(1) hummingbirdEM: Reads the input data, sets the initial values, executes the Expectation-Maximization (EM) algorithm, an estimation procedure for the Bayesian HMM and infers the best sequence of methylation states.

(2) hummingbirdPostAdjustment: Allows researchers to place extra requirements on the selection 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.

(3) hummingbirdGraph: Generates the Methylation Level and Prediction graphs for the user-specified regions.

Sample Dataset

A sample dataset "exampleHummingbird" is provided with the package as an example.

Specifically, partial data of chromosome 29 of the large offspring syndrome (LOS) study as described in "Chen, Z. et al (2017): Global misregulation of genes largely uncoupled to DNA methylome epimutations characterizes a congenital overgrowth syndrome. Scientific Reports 7, 12667". The raw FASTQ files of the WGBS experiment from this study are publibly available at Gene Expression Omnibus with accession no. GSE93775.

The matrices m_abnormUM, m_abnormM, m_normM, m_normUM and m_pos are given as an input to the functions of the hummingbird package. They are the methylation data from the WGBS experiment. In this experiment, there are four replicates in the case (LOS) group and four replicates in the control group. However, this is just an example. The hummingbird package does not require replication in either the control or the case group.

The objects hmmbird1 and hmmbird2 are the output of the hummingbirdEM and hummingbirdPostAdjustment functions, when the aforementioned matrices are used as described in the Example section.

Below is a short presentation of the data:

```
library(hummingbird)
data(exampleHummingbird)
```

The CpG positions:

```
m_pos[1:6,1]
```

```
## [1] 271 331 363 386 418 464
```

The matrices containing the methylated and unmethylated read count data of the normal group. Each column of the matrix represents a replicate and each row represents a CpG position:

```
m_normM[1:6,1:4]
```

```
##      [,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
```

```
m_normUM[1:6,1:4]
```

```
##      [,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
```

The matrices containing the methylated and unmethylated read count data of the abnormal group. Each column of the matrix represents a replicate and each row represents a CpG position:

```
m_abnormM[1:6,1:4]
```

```
##      [,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
```

```
m_abnormUM[1:6,1:4]
```

```
##      [,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
```

The output of the hummingbirdEM function:

```
str(hmmbird1)
```

```
## List of 2
##  $ obs       : int [1:3296, 1:4] 1 1 0 0 0 0 0 0 0 0 ...
##  $ normAbnorm: num [1:3296, 1:2] 0.672 0.258 0.156 0.122 0.333 ...
```

The output of the hummingbirdPostAdjustment function:

```
str(hmmbird2)
```

```
## List of 2
##  $ obsPostAdj: int [1:3296, 1:4] 0 0 0 0 0 0 0 0 0 0 ...
##  $ DMRs      : int [1:3, 1:6] 1 2 3 98391 107991 110551 98590 108350 110870 200 ...
```

Example

Loading the hummingbird package and the exampleHummingbird dataset (only the input matrices are required from the sample dataset):

```
library(hummingbird)
data(exampleHummingbird)
```

- hummingbirdEM: Expectation-Maximization Algorithm for Fitting the Hidden Markov Model

This function reads in methylated and unmethylated read count data, transforms it into logarithm bin-wise data, sets up initial values and implements the EM algorithm to estimate HMM parameters and find the best sequence of hidden states based on model fitting.

hmmbird1$obs is the output matrix containing the initial Direction, the Distance, the Start and End coordinates.

```
hmmbird1 <- hummingbirdEM(normM=m_normM, normUM=m_normUM, abnormM=m_abnormM,
abnormUM=m_abnormUM, pos=m_pos, binSize=40)
```

```
## Reading input...
## Bin size: 40.
## Total lines: 4746, total replicates: 4
## Processing input...
```

```
## Processing input completed...
## Calculation of the initial value...
## Initial Value calculated...
## EM begins...
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 10
## Iteration: 11
## Iteration: 12
## Iteration: 13
## Iteration: 14
## Iteration: 15
## Iteration: 16
## Iteration: 17
## Iteration: 18
## Iteration: 19
## Iteration: 20
## Iteration: 21
## Iteration: 22
## Calculation of states...
## EM converged after 22 iterations.
## Saving output...
## ****** Program ended. ******
```

- hummingbirdPostAdjustment: Post Adjustment algorithm for the output of the EM

As input to the function, the output matrix obs of the hummingbirdEM and the m_pos matrix is given.

This function adjusts HMM output such that each detected DMR has a (user-defined) minimum length, maximum gap and minimum number of CpGs in each DMR.

hmmbird2$DMRs is the output matrix containing the detected regions based on the user-defined arguments. The columns of the DMRs matrix are: Region number, Start genomic position, End genomic position, Length of region, Direction ("0" indicates no significant change, "1" indicates predicted hyper-methylation, and "2" indicates predicted hypo-methylation) and Number of CpGs.

```
hmmbird2 <- hummingbirdPostAdjustment(em=hmmbird1$obs, pos=m_pos, minCpGs=10,
minLength=100, maxGap=300)
```

```
## Reading input...
## Min CpGs: 10, Min Length: 100, Max gap: 300.
## Post Adjustment begins...
## Post Adjustment completed...
## Output DMRs...
## There are 3 DMRs in total. The first 3 are displayed.
## Region: Start, End, Length, Direction, CpGs
## 1: 98391, 98590, 200, 2, 10
```

```
## 2: 107991, 108350, 360, 2, 12
## 3: 110551, 110870, 320, 2, 10
## Saving output...
## ****** Program ended. ******
```

```
hmmbird2$DMRs
```

```
##      [,1]   [,2]   [,3] [,4] [,5] [,6]
## [1,]    1  98391  98590  200    2   10
## [2,]    2 107991 108350  360    2   12
## [3,]    3 110551 110870  320    2   10
```

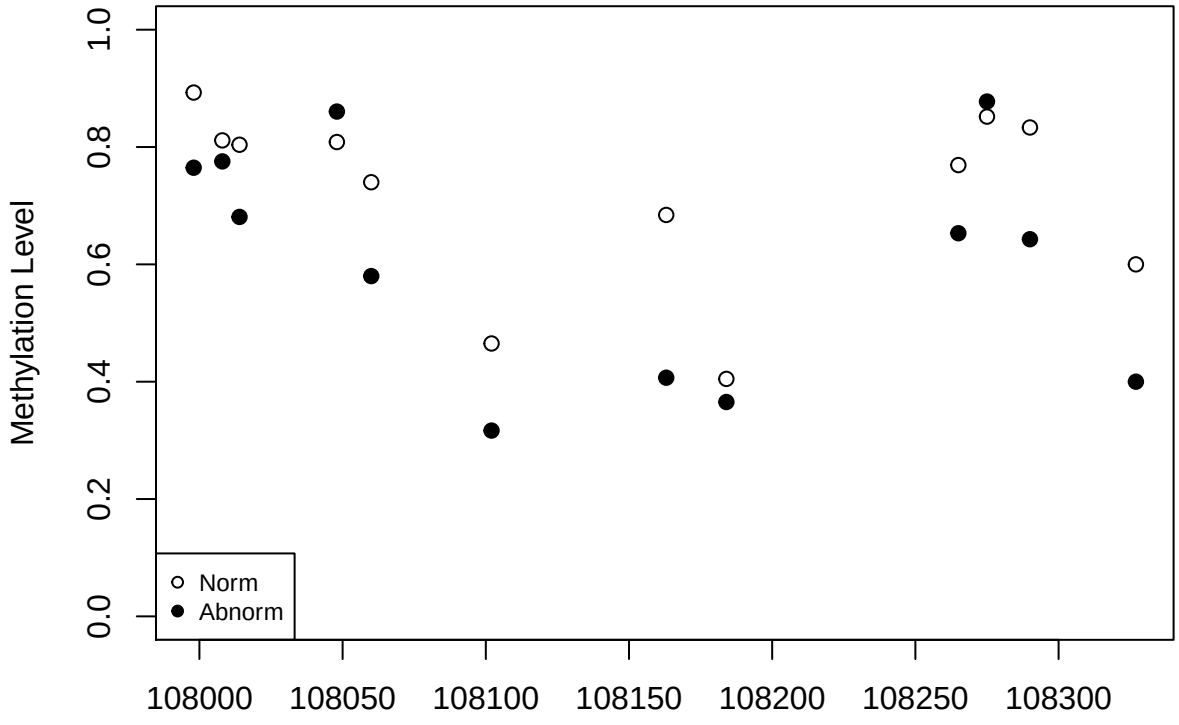
- hummingbirdGraph: Generates the Methylation Level and Prediction graphs for the user-specified regions.

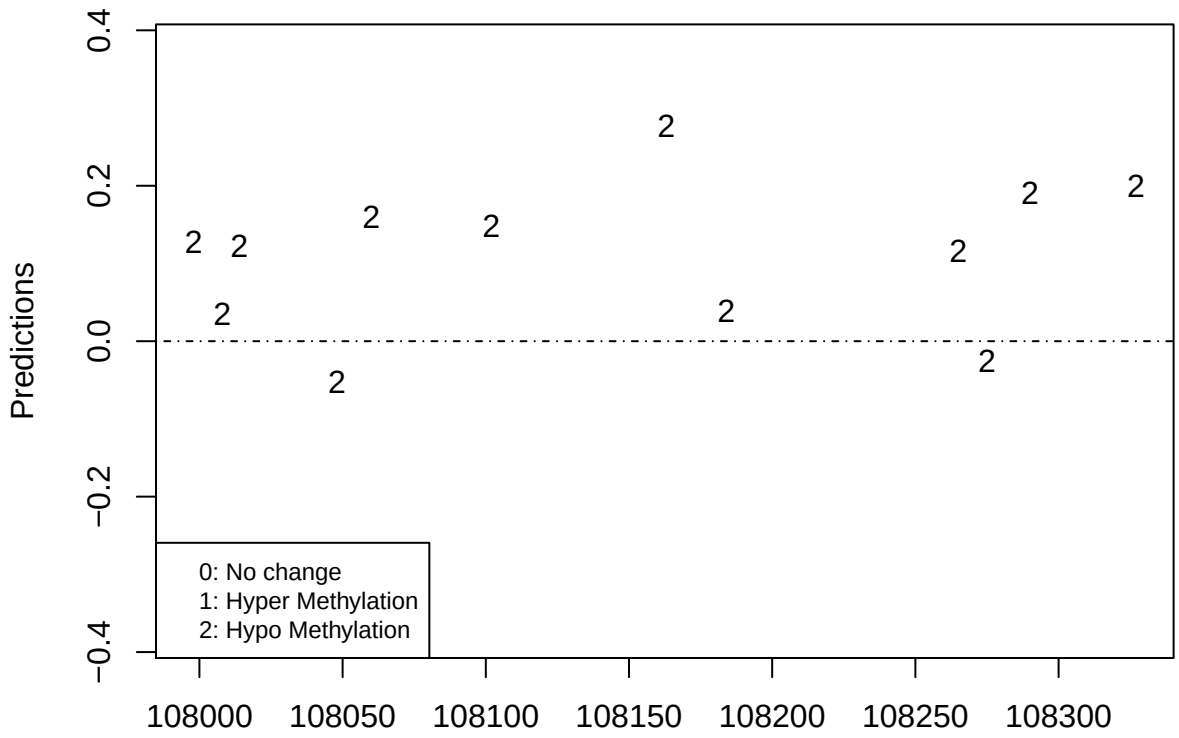
The Methylation Level graph is the sample averages from two comparison groups.

The Predictions graph is the sample average difference along with predictions, where "0" indicates no significant change, "1" indicates predicted hyper-methylation, and "2" indicates predicted hypo-methylation.

In the next figures, in order to visualize the 2nd DMR, which contains the largest number of CpGs (12), its coordinates (as specified in Start/End of hmmbird2$DMRs) are entered by the user:

```
hummingbirdGraph(pos=m_pos, normM=m_normM, normUM=m_normUM, abnormM=m_abnormM,
abnormUM=m_abnormUM, dmrs=hmmbird2$DMRs, coord1=107991, coord2=108350)
```





Citation

If you use the hummingbird package, please cite the following paper:

- Ji T. A Bayesian hidden Markov model for detecting differentially methylated regions. Biometrics. 2019;75(2):663-673. DOI:10.1111/biom.13000

The paper includes the detailed information of the algorithm.