

# Package ‘DMCHMM’

March 18, 2025

**Type** Package

**Title** Differentially Methylated CpG using Hidden Markov Model

**Version** 1.29.0

**Author** Farhad Shokoohi

**Maintainer** Farhad Shokoohi <shokoohi@icloud.com>

**Description** A pipeline for identifying differentially methylated CpG sites using Hidden Markov Model in bisulfite sequencing data. DNA methylation studies have enabled researchers to understand methylation patterns and their regulatory roles in biological processes and disease. However, only a limited number of statistical approaches have been developed to provide formal quantitative analysis. Specifically, a few available methods do identify differentially methylated CpG (DMC) sites or regions (DMR), but they suffer from limitations that arise mostly due to challenges inherent in bisulfite sequencing data. These challenges include: (1) that read-depths vary considerably among genomic positions and are often low; (2) both methylation and autocorrelation patterns change as regions change; and (3) CpG sites are distributed unevenly. Furthermore, there are several methodological limitations: almost none of these tools is capable of comparing multiple groups and/or working with missing values, and only a few allow continuous or multiple covariates. The last of these is of great interest among researchers, as the goal is often to find which regions of the genome are associated with several exposures and traits. To tackle these issues, we have developed an efficient DMC identification method based on Hidden Markov Models (HMMs) called “DMCHMM” which is a three-step approach (model selection, prediction, testing) aiming to address the aforementioned drawbacks.

**Depends** R (>= 4.1.0), SummarizedExperiment, methods, S4Vectors, BiocParallel, GenomicRanges, IRanges, fdrtool

**Imports** utils, stats, grDevices, rtracklayer, multcomp, calibrate, graphics

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**biocViews** DifferentialMethylation, Sequencing, HiddenMarkovModel, Coverage

**License** GPL-3  
**Date** 2020-09-27  
**Encoding** UTF-8  
**LazyData** true  
  
**BugReports** <https://github.com/shokoohi/DMCHMM/issues>  
**RoxygenNote** 7.1.2  
**NeedsCompilation** no  
**git\_url** <https://git.bioconductor.org/packages/DMCHMM>  
**git\_branch** devel  
**git\_last\_commit** d270747  
**git\_last\_commit\_date** 2024-10-29  
**Repository** Bioconductor 3.21  
**Date/Publication** 2025-03-18

## Contents

DMCHMM-package . . . . .	3
BSDData-class . . . . .	3
BSDMCs-class . . . . .	4
cBSDData-method . . . . .	5
cBSDMCs-method . . . . .	7
combine-method . . . . .	9
data . . . . .	10
findDMCs-method . . . . .	10
manhattanDMCs-method . . . . .	12
methHMEM-method . . . . .	13
methHMMCMC-method . . . . .	14
methLevels-method . . . . .	15
methReads-method . . . . .	16
methStates-method . . . . .	18
methVars-method . . . . .	19
params . . . . .	20
qqDMCs-method . . . . .	22
readBismark-method . . . . .	23
totalReads-method . . . . .	24
writeBED-method . . . . .	25

<b>Index</b>	<b>27</b>
--------------	-----------

DMCHMM-package

*Differentially Methylated CpG using Hidden Markov Model***Description**

DMCHMM is a novel profiling tool for identifying differentially methylated CpG sites using Hidden Markov Model in bisulfite sequencing data.

**DMCHMM methods**

[cBSData](#), [cBSDMCs](#), [methHMEM](#), [methHMMCMC](#), [findDMCs](#), [qqDMCs](#), [manhattanDMCs](#), [readBismark](#), [writeBED](#).

**DMCHMM objects**

[BSData-class](#), [BSDMCs-class](#)

BSData-class

*BSData object***Description**

The BSData object is an S4 class that represents BS-Seq Data.

**Arguments**

<code>methReads</code>	The matrix <code>methReads</code> contains the number of methylated reads spanning a CpG-site. The rows represent the CpG sites in <code>rowRanges</code> and the columns represent the samples in <code>colData</code> .
<code>totalReads</code>	The matrix <code>totalReads</code> contains the number of reads spanning a CpG-site. The rows represent the CpG sites in <code>rowRanges</code> and the columns represent the samples in <code>colData</code> .

**Value**

A [BSData-class](#) object

**Slots**

<code>methReads</code>	An integer matrix
<code>totalReads</code>	An integer matrix

**Author(s)**

Farhad Shokoohi <[shokoohi@icloud.com](mailto:shokoohi@icloud.com)>

**See Also**

[SummarizedExperiment](#) objects.

**Examples**

```
nr <- 500; nc <- 16
metht<-matrix(as.integer(runif(nr * nc, 0, nr)), nr)
methc<-matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
r1 <- GRanges(rep("chr1", nr), IRanges(1:nr, width=1), strand="*")
names(r1) <- 1:nr
cd1<-DataFrame(Group=rep(c("G1", "G2"),each=nc/2),row.names=LETTERS[1:nc])
OBJ1<-cBSDData(rowRanges=r1,methReads=methc,totalReads=metht,colData=cd1)
OBJ1
```

---

BSDMCs-class

*BSDMCs object*

---

**Description**

The BSDMCs object is an S4 class that represents differentially methylated CpG sites (DMCs) in BS-Seq Data.

**Arguments**

methReads	The matrix methReads contains the number of methylated reads spanning a CpG-site. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.
totalReads	The matrix totalReads contains the number of reads spanning a CpG-site. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.
methLevels	The matrix methLevels contains the predicted methylation level spanning a CpG-site using Hidden Markov model. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.
methStates	The matrix methStates contains the state of methylation obtained from Hidden Markov model spanning a CpG-site. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData. The value of state is stored in metadata, named Beta.
methVars	The matrix methVars contains the variances of the corresponding methLevels obtained from MCMC.

**Value**

A [BSDMCs-class](#) object

**Slots**

methReads An integer matrix  
 totalReads An integer matrix  
 methLevels A numeric matrix  
 methStates An integer matrix  
 methVars A double matrix

**Author(s)**

Farhad Shokoohi <shokoohi@icloud.com>

**Examples**

```
nr <- 500; nc <- 16
metht <- matrix(as.integer(runif(nr * nc, 0, nr)), nr)
methc <- matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
meths <- matrix(as.integer(runif(nr * nc, 0, 10)), nr)
methl <- methc/metht
methv <- matrix((runif(nr * nc, 0.1, 0.5)), nr)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1', 'G2'),each=nc/2),row.names=LETTERS[1:nc])
OBJ2 <- cBSDMCs(rowRanges=r1,methReads=methc,totalReads=metht,
methLevels=methl,methStates=meths,methVars=methv,colData=cd1)
OBJ2
```

---

cBSDData-method

*cBSDData method*


---

**Description**

Creates a [BSDData-class](#) object

**Usage**

```
cBSDData(
  methReads,
  totalReads,
  rowRanges,
  colData = DataFrame(row.names = colnames(methReads)),
  metadata = list(),
  ...
)

## S4 method for signature 'matrix,matrix,GRanges'
cBSDData(
  methReads,
```

```

    totalReads,
    rowRanges,
    colData = DataFrame(row.names = colnames(methReads)),
    metadata = list(),
    ...
  )

```

### Arguments

methReads	The matrix methReads contains the number of methylated reads spanning a CpG-site. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.
totalReads	The matrix totalReads contains the number of reads spanning a CpG-site. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.
rowRanges	A <a href="#">GRanges</a> or <a href="#">GRangesList</a> object describing the ranges of interest. Names, if present, become the row names of the <a href="#">SummarizedExperiment</a> object. The length of the <a href="#">GRanges</a> or <a href="#">GRangesList</a> must equal the number of rows of the matrices in assays. If rowRanges is missing, a <a href="#">SummarizedExperiment</a> instance is returned.
colData	Object of class "DataFrame" containing information on variable values of the samples
metadata	An optional list of arbitrary content describing the overall experiment
...	other possible parameters

### Details

The rows of a BSData object represent ranges (in genomic coordinates) of interest. The ranges of interest are described by a [GRanges](#) or a [GRangesList](#) object, accessible using the rowRanges function. The [GRanges](#) and [GRangesList](#) classes contains sequence (e.g., chromosome) name, genomic coordinates, and strand information. Each range can be annotated with additional data; this data might be used to describe the range or to summarize results (e.g., statistics of differential abundance) relevant to the range. Rows may or may not have row names; they often will not.

### Value

A [BSData-class](#) object

### Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

### Examples

```

nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')

```

```

names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1', 'G2'), each=nc/2), row.names=LETTERS[1:nc])
OBJ1 <- cBSDData(rowRanges=r1, methReads=methc, totalReads=methT, colData=cd1)
OBJ1

```

---

cBSDMCs-method

*cBSDMCs method*


---

## Description

Creates a [BSDMCs-class](#) object

## Usage

```

cBSDMCs(
  methReads,
  totalReads,
  methLevels,
  methStates,
  methVars,
  rowRanges,
  colData = DataFrame(row.names = colnames(methReads)),
  metadata = list(),
  ...
)

## S4 method for signature 'matrix,matrix,matrix,matrix,matrix,GRanges'
cBSDMCs(
  methReads,
  totalReads,
  methLevels,
  methStates,
  methVars,
  rowRanges,
  colData = DataFrame(row.names = colnames(methReads)),
  metadata = list(),
  ...
)

```

## Arguments

methReads	The matrix methReads contains the number of methylated reads spanning a CpG-site. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.
totalReads	The matrix totalReads contains the number of reads spanning a CpG-site. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.

methLevels	The matrix <code>methLevels</code> contains the predicted methylation level spanning a CpG-site using Hidden Markov model. The rows represent the CpG sites in <code>rowRanges</code> and the columns represent the samples in <code>colData</code> .
methStates	The matrix <code>methStates</code> contains the state of methylation obtained from Hidden Markov model spanning a CpG-site. The rows represent the CpG sites in <code>rowRanges</code> and the columns represent the samples in <code>colData</code> . The value of state is stored in <code>metadata</code> , named <code>Beta</code> .
methVars	The matrix <code>methVars</code> contains the variances of the corresponding <code>methLevels</code> obtained from MCMC.
rowRanges	A <code>GRanges</code> or <code>GRangesList</code> object describing the ranges of interest. Names, if present, become the row names of the <code>SummarizedExperiment</code> object. The length of the <code>GRanges</code> or <code>GRangesList</code> must equal the number of rows of the matrices in assays. If <code>rowRanges</code> is missing, a <code>SummarizedExperiment</code> instance is returned.
colData	Object of class "DataFrame" containing information on variable values of the samples
metadata	An optional list of arbitrary content describing the overall experiment
...	other possible parameters

## Details

The rows of a `BSDMCs` object represent ranges (in genomic coordinates) of interest. The ranges of interest are described by a `GRanges` or a `GRangesList` object, accessible using the `rowRanges` function. The `GRanges` and `GRangesList` classes contains sequence (e.g., chromosome) name, genomic coordinates, and strand information. Each range can be annotated with additional data; this data might be used to describe the range or to summarize results (e.g., statistics of differential abundance) relevant to the range. Rows may or may not have row names; they often will not.

## Value

A `BSDMCs-class`

## Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

## Examples

```
set.seed(1980)
nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
meths <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methl <- methc/metht
methv <- matrix((runif(nr * nc, 0.1, 0.5)), nr)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1', 'G2'),each=nc/2),row.names=LETTERS[1:nc])
```



```
OBJ2 <- cBDMCs(rowRanges=r1, methReads=methc, totalReads=metht,
methLevels=methl, methStates=meths, methVars=methv, colData=cd1)
OBJ2
```

---

combine-method	<i>combine method</i>
----------------	-----------------------

---

## Description

combine two [BSDData-class](#) or two [BDMCs-class](#)

## Usage

```
combine(obj1, obj2)

## S4 method for signature 'BSDData, BSDData'
combine(obj1, obj2)

## S4 method for signature 'BDMCs, BDMCs'
combine(obj1, obj2)
```

## Arguments

```
obj1          A BSDData-class or BDMCs-class
obj2          A BSDData-class or BDMCs-class
```

## Value

A [BSDData-class](#) or [BDMCs-class](#)

## Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

## Examples

```
set.seed(1980)
nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc*2, 0, nr)), nr)
methc <- matrix(rbinom(n=nr*nc, c(metht), prob = runif(nr*nc*2)), nr, nc*2)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep('G1', each=nc), row.names=LETTERS[1:nc])
OBJ1 <- cBSDData(rowRanges=r1, methReads=methc[, 1:nc], totalReads=metht[, 1:nc],
colData=cd1)
cd2 <- DataFrame(Group=rep('G2', each=nc), row.names=LETTERS[nc+1:nc])
OBJ2 <- cBSDData(rowRanges=r1, methReads=methc[, nc+1:nc], totalReads=
metht[, nc+1:nc], colData=cd2)
OBJ3 <- combine(OBJ1, OBJ2)
OBJ3
```

---

data

*data*

---

**Description**

A part of BS-Seq data for three cell type: WGBS data were derived from whole blood collected on a cohort of healthy individuals from Sweden. Cell lines were separated into T-cells (19 samples), monocytes (13 samples) and B-cells (8 samples). Sequencing was performed on the Illumina HiSeq2000/2500 system for each of the 40 samples, separately. For illustration only 3 samples each containing 30,440 CpG sites around BLK gene are provided here. The whole data are analyzed in the cited paper.

**Format**

BED files

**Details**

The data is part of whole blood from Sweden.

**Author(s)**

Farhad Shokoohi <shokoohi@icloud.com>

**Source**

Genomic Quebec

---

findDMCs-method

*findDMCs method*

---

**Description**

finds the DMCs after smoothing using HMM

**Usage**

```
findDMCs(  
  object,  
  formula,  
  FDRthreshold,  
  Methylthreshold,  
  mc.cores,  
  windowsize,  
  weightfunction  
)
```

```
## S4 method for signature 'BSDMCs'
findDMCs(
  object,
  formula,
  FDRthreshold,
  Methylthreshold,
  mc.cores,
  windowsize,
  weightfunction
)
```

### Arguments

object            A [BSDData-class](#) or [BSDMCs-class](#) object

formula            A formula

FDRthreshold      A numeric value

Methylthreshold                    A positive numeric value; the default is 0.001

mc.cores            An integer greater than 0

windowsize        An integer value for partitioning data into windows of size windowsize.

weightfunction    A function to create weights using variance obtained from the MCMC algorithm

### Value

[BSDMCs-class](#) object

### Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

### Examples

```
set.seed(1980)
nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1', 'G2'), each=nc/2), row.names=LETTERS[1:nc])
OBJ1 <- cBSDData(rowRanges=r1, methReads=methc, totalReads=metht, colData=cd1)
OBJ2 <- methHMEM(OBJ1, MaxK=2, mc.cores=2)
OBJ3 <- methHMMCMC(OBJ2, mc.cores=2)
OBJ4 <- findDMCs(OBJ3, mc.cores=2)
head(metadata(OBJ4)$DMCHMM)
```

---

manhattanDMCs-method *manhattanDMCs method*

---

### Description

Creates a Manhattan plot based on the p-values obtained from [findDMCs](#) method

### Usage

```
manhattanDMCs(  
  object,  
  col,  
  chrlabs,  
  suggestiveline,  
  genomewideline,  
  highlight,  
  logp,  
  annotatePval,  
  annotateTop,  
  ...  
)  
  
## S4 method for signature 'BSDMCs'  
manhattanDMCs(  
  object,  
  col,  
  chrlabs,  
  suggestiveline,  
  genomewideline,  
  highlight,  
  logp,  
  annotatePval,  
  annotateTop,  
  ...  
)
```

### Arguments

object	A <a href="#">BSDData-class</a> or <a href="#">BSDMCs-class</a> object
col	A character vector indicating which colors to alternate.
chrlabs	A character vector equal to the number of chromosomes specifying the chromosome labels (e.g., <code>c(1:22, "X", "Y", "MT")</code> ).
suggestiveline	Where to draw a "suggestive" line. Default $-\log_{10}(1e-5)$ . Set to FALSE to disable.
genomewideline	Where to draw a "genome-wide significant" line. Default $-\log_{10}(5e-8)$ . Set to FALSE to disable.

highlight	A character vector of SNPs in your dataset to highlight. These SNPs should all be in your dataset.
logp	If TRUE, the $-\log_{10}$ of the p-value is plotted. It isn't very useful to plot raw p-values, but plotting the raw value could be useful for other genome-wide plots, for example, peak heights, bayes factors, test statistics, other "scores," etc.
annotatePval	If set, SNPs below this p-value will be annotated on the plot.
annotateTop	If TRUE, only annotates the top hit on each chromosome that is below the annotatePval threshold.
...	other possible parameters

**Value**

A Manhattan plot

**Author(s)**

Farhad Shokoohi <shokoohi@icloud.com>

**Examples**

```
set.seed(1980)
nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1', 'G2'), each=nc/2), row.names=LETTERS[1:nc])
OBJ1 <- cBSData(rowRanges=r1,methReads=methc,totalReads=metht,colData=cd1)
OBJ2 <- methHMEM(OBJ1, MaxK=2, mc.cores=2)
OBJ3 <- methHMMCMC(OBJ2, mc.cores=2)
OBJ4 <- findDMCs(OBJ3, mc.cores=2)
manhattanDMCs(OBJ4)
```

---

methHMEM-method

*methHMEM method*

---

**Description**

Estimates the HMM methylation paths and the HMM order for each sample using the EM algorithm

**Usage**

```
methHMEM(object, MaxK, MaxEmitter, epsEM, useweight, mc.cores)
```

```
## S4 method for signature 'BSData'
```

```
methHMEM(object, MaxK, MaxEmitter, epsEM, useweight, mc.cores)
```

**Arguments**

object	A <a href="#">BSDData-class</a> or <a href="#">BSDMCs-class</a> object
MaxK	An integer value
MaxEmitter	An integer value
epsEM	A positive numeric value
useweight	A logical value
mc.cores	An integer greater than 0

**Value**

[BSDMCs-class](#) object

**Author(s)**

Farhad Shokoohi <shokoohi@icloud.com>

**Examples**

```
set.seed(1980)
nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c=metht,prob = runif(nr*nc)),nr,nc)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1', 'G2'), each=nc/2), row.names=LETTERS[1:nc])
OBJ1 <- cBSDData(rowRanges=r1, methReads=methc, totalReads=metht, colData=cd1)
OBJ2 <- methHMEM(OBJ1, MaxK=2, mc.cores=2)
OBJ2
```

---

methHMMCMC-method

*methHMMCMC method*

---

**Description**

Estimates the HMM methylation paths and the HMM order for each sample using the MCMC algorithm

**Usage**

```
methHMMCMC(object, useweight, nburn, nthin, nsamp, mc.cores)
```

```
## S4 method for signature 'BSDMCs'
```

```
methHMMCMC(object, useweight, nburn, nthin, nsamp, mc.cores)
```

**Arguments**

object	A <a href="#">BSDData-class</a> or <a href="#">BSDMCs-class</a> object
useweight	A logical value
nburn	An integer value
nthin	An integer value
nsamp	An integer value
mc.cores	An integer greater than 0

**Value**

[BSDMCs-class](#) object

**Author(s)**

Farhad Shokoohi <shokoohi@icloud.com>

**Examples**

```
set.seed(1980)
nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c=metht),prob = runif(nr*nc),nr,nc)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1', 'G2'),each=nc/2),row.names=LETTERS[1:nc])
OBJ1 <- cBSDData(rowRanges=r1,methReads=methc,totalReads=metht,colData=cd1)
OBJ2 <- methHMEM(OBJ1, MaxK=2, mc.cores=2)
OBJ3 <- methHMMC(OBJ2, mc.cores=2)
OBJ3
```

---

methLevels-method      *methLevels method*

---

**Description**

Returns methLevels stored in [BSDMCs-class](#)

Assigns methLevels to [BSDMCs-class](#)

**Usage**

```
methLevels(object)

methLevels(object) <- value

## S4 method for signature 'BSDMCs'
methLevels(object)
```

```
## S4 replacement method for signature 'BSDMCs,matrix'
methLevels(object) <- value
```

### Arguments

object            A [BSDData-class](#) or [BSDMCs-class](#) object  
value             An integer matrix

### Value

A matrix  
A [BSDMCs-class](#) object

### Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

### Examples

```
set.seed(1980)
nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
meths <- matrix(as.integer(runif(nr * nc, 0, 10)), nr)
methl <- methc/metht
methv <- matrix((runif(nr * nc, 0.1, 0.5)), nr)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1', 'G2'),each=nc/2),row.names=LETTERS[1:nc])
OBJ2 <- cBSDMCs(rowRanges=r1,methReads=methc,totalReads=metht,
methLevels=methl,methStates=meths,methVars=methv,colData=cd1)
methLevels(OBJ2)
methLevels(OBJ2) <- methl
```

---

methReads-method            *methReads method*

---

### Description

Returns methReads stored in [BSDData-class](#)  
Assigns methReads to [BSDData-class](#)  
Returns methReads stored in [BSDMCs-class](#)  
Assigns methReads to [BSDMCs-class](#)



**Usage**

```

methReads(object)

methReads(object) <- value

methReads(object)

methReads(object) <- value

## S4 method for signature 'BSDData'
methReads(object)

## S4 replacement method for signature 'BSDData,matrix'
methReads(object) <- value

## S4 method for signature 'BSDMCs'
methReads(object)

## S4 replacement method for signature 'BSDMCs,matrix'
methReads(object) <- value

```

**Arguments**

object	A <a href="#">BSDData-class</a> or <a href="#">BSDMCs-class</a> object
value	An integer matrix

**Value**

A matrix  
A [BSDData-class](#) object  
A matrix  
A [BSDMCs-class](#) object

**Author(s)**

Farhad Shokoohi <shokoohi@icloud.com>

**Examples**

```

nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1','G2'),each=nc/2),row.names=LETTERS[1:nc])
OBJ1 <- cBSDData(rowRanges=r1,methReads=methc,totalReads=metht,colData=cd1)
methReads(OBJ1)
methReads(OBJ1) <- methc

```

---

methStates-method      *methStates method*

---

### Description

Returns methStates stored in [BSDMCs-class](#)

Assigns methStates to [BSDMCs-class](#)

### Usage

```
methStates(object)
```

```
methStates(object) <- value
```

```
## S4 method for signature 'BSDMCs'
```

```
methStates(object)
```

```
## S4 replacement method for signature 'BSDMCs,matrix'
```

```
methStates(object) <- value
```

### Arguments

object            A [BSDData-class](#) or [BSDMCs-class](#) object

value             An integer matrix

### Value

A matrix

A [BSDMCs-class](#) object

### Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

### Examples

```
set.seed(1980)
nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
meths <- matrix(as.integer(runif(nr * nc, 0, 10)), nr)
methl <- methc/metht
methv <- matrix((runif(nr * nc, 0.1, 0.5)), nr)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1', 'G2'),each=nc/2),row.names=LETTERS[1:nc])
OBJ2 <- cBSDMCs(rowRanges=r1,methReads=methc,totalReads=metht,
methLevels=methl,methStates=meths,methVars=methv,colData=cd1)
```

```
methStates(OBJ2)
methStates(OBJ2)<- meths
```

---

```
methVars-method      methVars method
```

---

## Description

Returns methVars stored in [BSDMCs-class](#)

Assigns methVars to [BSDMCs-class](#)

## Usage

```
methVars(object)

methVars(object) <- value

## S4 method for signature 'BSDMCs'
methVars(object)

## S4 replacement method for signature 'BSDMCs,matrix'
methVars(object) <- value
```

## Arguments

object	A <a href="#">BSDData-class</a> or <a href="#">BSDMCs-class</a> object
value	An integer matrix

## Value

A matrix  
 A [BSDMCs-class](#) object

## Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

## Examples

```
set.seed(1980)
nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
meths <- matrix(as.integer(runif(nr * nc, 0, 10)), nr)
methl <- methc/metht
methv <- matrix((runif(nr * nc, 0.1, 0.5)), nr)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
```

```

cd1 <- DataFrame(Group=rep(c('G1', 'G2'), each=nc/2), row.names=LETTERS[1:nc])
OBJ2 <- cBSDMCs(rowRanges=r1, methReads=methc, totalReads=metht,
methLevels=methl, methStates=meths, methVars=methv, colData=cd1)
methVars(OBJ2)
methVars(OBJ2)<- meths

```

---

 params

*params*


---

## Description

parameters name and their descriptions

## Arguments

methReads	The matrix methReads contains the number of methylated reads spanning a CpG-site. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.
totalReads	The matrix totalReads contains the number of reads spanning a CpG-site. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.
methLevels	The matrix methLevels contains the predicted methylation level spanning a CpG-site using Hidden Markov model. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.
methVars	The matrix methVars contains the variances of the corresponding methLevels obtained from MCMC.
methStates	The matrix methStates contains the state of methylation obtained from Hidden Markov model spanning a CpG-site. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData. The value of state is stored in metadata, named Beta.
rowRanges	A <a href="#">GRanges</a> or <a href="#">GRangesList</a> object describing the ranges of interest. Names, if present, become the row names of the <a href="#">SummarizedExperiment</a> object. The length of the <a href="#">GRanges</a> or <a href="#">GRangesList</a> must equal the number of rows of the matrices in assays. If rowRanges is missing, a <a href="#">SummarizedExperiment</a> instance is returned.
colData	Object of class "DataFrame" containing information on variable values of the samples
metadata	An optional list of arbitrary content describing the overall experiment
object	A <a href="#">BSDData-class</a> or <a href="#">BSDMCs-class</a> object
value	An integer matrix
obj1	A <a href="#">BSDData-class</a> or <a href="#">BSDMCs-class</a>
obj2	A <a href="#">BSDData-class</a> or <a href="#">BSDMCs-class</a>
files	A character list

file	A character
name	A character list
MaxK	An integer value
MaxEmitter	An integer value
epsEM	A positive numeric value
useweight	A logical value
mc.cores	An integer greater than 0
nburn	An integer value
nthin	An integer value
nsamp	An integer value
formula	A formula
FDRthreshold	A numeric value
Methylthreshold	A positive numeric value; the default is 0.001
weightfunction	A function to create weights using variance obtained from the MCMC algorithm
...	other possible parameters
col	A character vector indicating which colors to alternate.
chrlabs	A character vector equal to the number of chromosomes specifying the chromosome labels (e.g., c(1:22, "X", "Y", "MT")).
suggestiveline	Where to draw a "suggestive" line. Default $-\log_{10}(1e-5)$ . Set to FALSE to disable.
genomewideline	Where to draw a "genome-wide significant" line. Default $-\log_{10}(5e-8)$ . Set to FALSE to disable.
highlight	A character vector of SNPs in your dataset to highlight. These SNPs should all be in your dataset.
logp	If TRUE, the $-\log_{10}$ of the p-value is plotted. It isn't very useful to plot raw p-values, but plotting the raw value could be useful for other genome-wide plots, for example, peak heights, bayes factors, test statistics, other "scores," etc.
annotatePval	If set, SNPs below this p-value will be annotated on the plot.
annotateTop	If TRUE, only annotates the top hit on each chromosome that is below the annotatePval threshold.
windowSize	An integer value for partitioning data into windows of size windowSize.

**Author(s)**

Farhad Shokoohi <shokoohi@icloud.com>

---

`qqDMCs-method`*qqDMCs method*

---

**Description**

Creates a Q-Q plot based on the p-values obtained from `findDMCs` method

**Usage**

```
qqDMCs(object, ...)
```

```
## S4 method for signature 'BSDMCs'  
qqDMCs(object, ...)
```

**Arguments**

```
object      A BSDData-class or BSDMCs-class object  
...         other possible parameters
```

**Value**

A QQ plot

**Author(s)**

Farhad Shokoohi <[shokoohi@icloud.com](mailto:shokoohi@icloud.com)>

**Examples**

```
set.seed(1980)  
nr <- 150; nc <- 8  
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)  
methc <- matrix(rbinom(n=nr*nc,c=metht,prob = runif(nr*nc)),nr,nc)  
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')  
names(r1) <- 1:nr  
cd1 <- DataFrame(Group=rep(c('G1', 'G2'),each=nc/2),row.names=LETTERS[1:nc])  
OBJ1 <- cBSDData(rowRanges=r1,methReads=methc,totalReads=metht,colData=cd1)  
OBJ2 <- methHMEM(OBJ1, MaxK=2, mc.cores=2)  
OBJ3 <- methHMMC(OBJ2, mc.cores=2)  
OBJ4 <- findDMCs(OBJ3, mc.cores=2)  
qqDMCs(OBJ4)
```

---

readBismark-method      *readBismark method*

---

## Description

reads BS-Seq data

## Usage

```
readBismark(files, colData, mc.cores)

## S4 method for signature 'character,DataFrame,numeric'
readBismark(files, colData, mc.cores)

## S4 method for signature 'character,data.frame,numeric'
readBismark(files, colData, mc.cores)

## S4 method for signature 'character,character,numeric'
readBismark(files, colData, mc.cores)
```

## Arguments

files	A character list
colData	Object of class "DataFrame" containing information on variable values of the samples
mc.cores	An integer greater than 0

## Value

A [BSData-class](#) object

## Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

## Examples

```
fn <- list.files(system.file('extdata',package = 'DMCHMM'))
fn.f <- list.files(system.file('extdata',package='DMCHMM'), full.names=TRUE)
OBJ <- readBismark(fn.f, fn, mc.cores=2)
cdOBJ <- DataFrame(Cell = factor(c('BC', 'TC', 'Mono'),
labels = c('BC', 'TC', 'Mono')), row.names = c('BCU1568', 'BCU173', 'BCU551'))
colData(OBJ) <- cdOBJ
OBJ
```

---

totalReads-method	<i>totalReads method</i>
-------------------	--------------------------

---

### Description

Returns totalReads stored in [BSDData-class](#)

Assigns totalReads to [BSDData-class](#)

Returns totalReads stored in [BSDMCs-class](#)

Assigns totalReads to [BSDMCs-class](#)

### Usage

```
totalReads(object)
```

```
totalReads(object) <- value
```

```
totalReads(object)
```

```
totalReads(object) <- value
```

```
## S4 method for signature 'BSDData'  
totalReads(object)
```

```
## S4 replacement method for signature 'BSDData,matrix'  
totalReads(object) <- value
```

```
## S4 method for signature 'BSDMCs'  
totalReads(object)
```

```
## S4 replacement method for signature 'BSDMCs,matrix'  
totalReads(object) <- value
```

### Arguments

object            A [BSDData-class](#) or [BSDMCs-class](#) object

value            An integer matrix

### Value

A matrix

A [BSDData-class](#) object

A matrix

A [BSDMCs-class](#) object



**Author(s)**

Farhad Shokoohi <shokoohi@icloud.com>

**Examples**

```
nr <- 150; nc <- 8
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n=nr*nc,c(metht),prob = runif(nr*nc)),nr,nc)
r1 <- GRanges(rep('chr1', nr), IRanges(1:nr, width=1), strand='*')
names(r1) <- 1:nr
cd1 <- DataFrame(Group=rep(c('G1', 'G2'),each=nc/2),row.names=LETTERS[1:nc])
OBJ1 <- cBSDData(rowRanges=r1,methReads=methc,totalReads=metht,colData=cd1)
totalReads(OBJ1)
totalReads(OBJ1) <- metht
```

---

writeBED-method

*writeBED method*

---

**Description**

write BS-Seq data to BED files

**Usage**

```
writeBED(object, name, file)

## S4 method for signature 'BSDData,character,character'
writeBED(object, name, file)

## S4 method for signature 'BSDData,character,missing'
writeBED(object, name)

## S4 method for signature 'BSDData,missing,character'
writeBED(object, file)

## S4 method for signature 'BSDData,missing,missing'
writeBED(object)

## S4 method for signature 'BSDMCs,character,character'
writeBED(object, name, file)

## S4 method for signature 'BSDMCs,character,missing'
writeBED(object, name)

## S4 method for signature 'BSDMCs,missing,character'
writeBED(object, file)

## S4 method for signature 'BSDMCs,missing,missing'
writeBED(object)
```

**Arguments**

object	A <a href="#">BSDData-class</a> or <a href="#">BSDMCs-class</a> object
name	A character list
file	A character

**Value**

BED files

**Author(s)**

Farhad Shokoohi <[shokoohi@icloud.com](mailto:shokoohi@icloud.com)>

# Index

- \* **manhattan**
  - manhattanDMCs-method, 12
- \* **object**
  - BSDData-class, 3
  - BSDMCs-class, 4
- \* **visualization**
  - manhattanDMCs-method, 12
- BSDData (BSDData-class), 3
- BSDData-class, 3
- BSDMCs (BSDMCs-class), 4
- BSDMCs-class, 4
- cBSDData, 3
- cBSDData (cBSDData-method), 5
- cBSDData, matrix, matrix, GRanges-method (cBSDData-method), 5
- cBSDData-method, 5
- cBSDMCs, 3
- cBSDMCs (cBSDMCs-method), 7
- cBSDMCs, matrix, matrix, matrix, matrix, matrix, GRanges-method (cBSDMCs-method), 7
- cBSDMCs-method, 7
- combine (combine-method), 9
- combine, BSDData, BSDData-method (combine-method), 9
- combine, BSDMCs, BSDMCs-method (combine-method), 9
- combine-method, 9
- data, 10
- DMCHMM (DMCHMM-package), 3
- DMCHMM-package, 3
- findDMCs, 3, 12, 22
- findDMCs (findDMCs-method), 10
- findDMCs, BSDMCs-method (findDMCs-method), 10
- findDMCs-method, 10
- GRanges, 6, 8, 20
- GRangesList, 6, 8, 20
- manhattanDMCs, 3
- manhattanDMCs (manhattanDMCs-method), 12
- manhattanDMCs, BSDMCs-method (manhattanDMCs-method), 12
- manhattanDMCs-method, 12
- methHMEM, 3
- methHMEM (methHMEM-method), 13
- methHMEM, BSDData-method (methHMEM-method), 13
- methHMEM-method, 13
- methHMMCMC, 3
- methHMMCMC (methHMMCMC-method), 14
- methHMMCMC, BSDMCs-method (methHMMCMC-method), 14
- methHMMCMC-method, 14
- methLevels (methLevels-method), 15
- methLevels, BSDMCs-method (methLevels-method), 15
- methLevels-method, 15
- methLevels<- (methLevels-method), 15
- methLevels<-, BSDMCs, matrix-method (methLevels-method), 15
- methReads (methReads-method), 16
- methReads, BSDData-method (methReads-method), 16
- methReads, BSDMCs-method (methReads-method), 16
- methReads-method, 16
- methReads<- (methReads-method), 16
- methReads<-, BSDData, matrix-method (methReads-method), 16
- methReads<-, BSDMCs, matrix-method (methReads-method), 16
- methStates (methStates-method), 18
- methStates, BSDMCs-method (methStates-method), 18
- methStates-method, 18
- methStates<- (methStates-method), 18

- methStates<- ,BSDMCs,matrix-method  
(methStates-method), 18
- methVars (methVars-method), 19
- methVars,BSDMCs-method  
(methVars-method), 19
- methVars-method, 19
- methVars<- (methVars-method), 19
- methVars<- ,BSDMCs,matrix-method  
(methVars-method), 19
  
- params, 20
  
- qqDMCs, 3
- qqDMCs (qqDMCs-method), 22
- qqDMCs,BSDMCs-method (qqDMCs-method), 22
- qqDMCs-method, 22
  
- readBismark, 3
- readBismark (readBismark-method), 23
- readBismark,character,character,numeric-method  
(readBismark-method), 23
- readBismark,character,data.frame,numeric-method  
(readBismark-method), 23
- readBismark,character,DataFrame,numeric-method  
(readBismark-method), 23
- readBismark-method, 23
  
- SummarizedExperiment, 4, 6, 8, 20
  
- totalReads (totalReads-method), 24
- totalReads,BSDData-method  
(totalReads-method), 24
- totalReads,BSDMCs-method  
(totalReads-method), 24
- totalReads-method, 24
- totalReads<- (totalReads-method), 24
- totalReads<- ,BSDData,matrix-method  
(totalReads-method), 24
- totalReads<- ,BSDMCs,matrix-method  
(totalReads-method), 24
  
- writeBED, 3
- writeBED (writeBED-method), 25
- writeBED,BSDData,character,character-method  
(writeBED-method), 25
- writeBED,BSDData,character,missing-method  
(writeBED-method), 25
- writeBED,BSDData,missing,character-method  
(writeBED-method), 25
- writeBED,BSDData,missing,missing-method  
(writeBED-method), 25
- writeBED,BSDMCs,character,character-method  
(writeBED-method), 25
- writeBED,BSDMCs,character,missing-method  
(writeBED-method), 25
- writeBED,BSDMCs,missing,character-method  
(writeBED-method), 25
- writeBED,BSDMCs,missing,missing-method  
(writeBED-method), 25
- writeBED-method, 25