Package 'DMCFB'

November 11, 2026

Type Package

Title Differentially Methylated Cytosines via a Bayesian Functional Approach

Version 1.26.0

Description DMCFB is a pipeline for identifying differentially methylated cytosines using a Bayesian functional regression model in bisulfite sequencing data. By using a functional regression data model, it tries to capture position-specific, group-specific and other covariates-specific methylation patterns as well as spatial correlation patterns and unknown underlying models of methylation data. It is robust and flexible with respect to the true underlying models and inclusion of any covariates, and the missing values are imputed using spatial correlation between positions and samples. A Bayesian approach is adopted for estimation and inference in the proposed method.

Depends R (>= 4.0.0), SummarizedExperiment, methods, S4Vectors, BiocParallel, GenomicRanges, IRanges

Imports utils, stats, speedglm, MASS, data.table, splines, arm, rtracklayer, benchmarkme, tibble, matrixStats, fastDummies, graphics

Suggests testthat, knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

biocViews DifferentialMethylation, Sequencing, Coverage, Bayesian, Regression

License GPL-3

Encoding UTF-8

LazyData true

BugReports https://github.com/shokoohi/DMCFB/issues

RoxygenNote 7.3.0

git_url https://git.bioconductor.org/packages/DMCFB

git_branch RELEASE_3_23

git_last_commit b584dc0

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Farhad Shokoohi [aut, cre] (ORCID:
<https://orcid.org/0000-0002-6224-2609>)

Maintainer Farhad Shokoohi <shokoohi@icloud.com>

Contents

DMCFB-package 2
BSDMC-class 3
cBSDMC-method 4
combine-method 6
findDMCFB-method 7
methLevels-method 9
methReads-method 10
params 11
plotDMCFB-method 13
readBismark-method 14
totalReads-method 15

Index **17**

`DMCFB-package` *Differentially Methylated cytosines using functional Bayesian regression models*

Description

DMCFB is a profiling tool for identifying differentially methylated cytosines using Functional Bayesian Model in bisulfite sequencing data.

`DMCFB` methods

`findDMCFB`, `plotDMCFB`, `cBSDMC`, `readBismark`.

`BSDMC` objects

`BSDMC-class`

BSDMC-class *BSDMC object*

Description

The BSDMC 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 Bayesian functional regression models. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.

Value

A BSDMC-class object

Slots

methReads An integer matrix

totalReads An integer matrix

methLevels A numeric matrix

Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

See Also

RangedSummarizedExperiment-class GRanges-class

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

---

cBSDMC-method *cBSDMC method*

Description

Creates a BSDMC-class object

Usage

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

## S4 method for signature 'matrix,matrix,matrix,GRanges'
cBSDMC(
  methReads,
  totalReads,
  methLevels,
  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 methLevels contains the predicted methylation level spanning a CpG-site using Bayesian functional regression models. The rows represent the CpG sites in rowRanges and the columns represent the samples in colData.
rowRanges	A GRanges or GRangesList object describing the ranges of interest. Names, if present, become the row names of the SummarizedExperiment object. The length of the GRanges or GRangesList must equal the number of rows of the matrices in assays. If rowRanges is missing, a SummarizedExperiment instance is returned.
colData	Object of class 'DataFrame' containing information on variable values of the samples
metadata	A list of storing MCMC samples or DMCs
...	other possible parameters

Details

The rows of a BSDMC 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 BSDMC-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, 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 <- cBSDMC(
  rowRanges = r1, methReads = methc, totalReads = metht,
  methLevels = methl, methStates = meths, methVars = methv, colData = cd1
)
```

OBJ2

---

combine-method *combine method*

Description

combine two BSDMC-class or two BSDMC-class

Usage

```
combine(obj1, obj2)

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

Arguments

obj1	A BSDMC-class
obj2	A BSDMC-class

Value

A BSDMC-class or BSDMC-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
)
methl <- methc / metht
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 <- cBSDMC(
  rowRanges = r1, methReads = methc[, 1:nc], totalReads = metht[, 1:nc],
  methLevels = methl[, 1:nc], colData = cd1
)
cd2 <- DataFrame(
  Group = rep("G2", each = nc),
```

```
  row.names = LETTERS[nc + 1:nc]
)
OBJ2 <- cBSDMC(
  rowRanges = r1, methReads = methc[, nc + 1:nc], totalReads =
    metht[, nc + 1:nc], methLevels = methl[, nc + 1:nc], colData = cd2
)
OBJ3 <- combine(OBJ1, OBJ2)
OBJ3
```

---

findDMCFB-method *findDMCFB method*

Description

DMC identification via Bayesian functional regression models

Usage

```
findDMCFB(
  object,
  bwa,
  bwb,
  nBurn,
  nMC,
  nThin,
  alpha,
  sdv,
  nCores,
  pSize,
  sfiles
)

## S4 method for signature 'BSDMC'
findDMCFB(
  object,
  bwa,
  bwb,
  nBurn,
  nMC,
  nThin,
  alpha,
  sdv,
  nCores,
  pSize,
  sfiles
)
```

Arguments

object	A BSDMC-class object
bwa	An integer value specifying the band-width size of B-spline basis matrix for a natural cubic spline for the group-specific effects of the Bayesian functional regression model
bwb	An integer value specifying the band-width size of B-spline basis matrix for a natural cubic spline for the individual-specific effects of the Bayesian functional regression model
nBurn	An integer value specifying the number of burn-in samples
nMC	An integer value specifying the number of MCMC samples after burn-in
nThin	An integer value specifying the thining number in MCMC
alpha	A numeric value specifying the level of α in credible interval $(1-\alpha)\%$
sdv	An double value specifying the standard deviation of priors
nCores	An integer value specifying the number of machine cores for parallel computing
pSize	An integer value specifying the number of cytosines in a regrion to be used in a Bayesian functiona regression model for DMC detection
sfiles	A logical value indicating whether files to be saved or not.

Value

BSDMC-class object

Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

Examples

```
set.seed(1980)
nr <- 1000
nc <- 4
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n = nr * nc, c(metht), prob = runif(nr * nc)), nr, nc)
methl <- methc / metht
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 <- cBSDMC(
  rowRanges = r1, methReads = methc, totalReads = metht,
  methLevels = methl, colData = cd1
)
OBJ2 <- findDMCFB(OBJ1,
  bwa = 10, bwb = 10, nBurn = 50, nMC = 50, nThin = 1,
  alpha = 0.05, nCores = 2, pSize = 500, sfiles = FALSE
```

```
)
OBJ2
```

---

methLevels-method *methLevels method*

Description

Returns methLevels stored in BSDMC-class

Assigns methLevels to BSDMC-class

Usage

```
methLevels(object)

methLevels(object) <- value

## S4 method for signature 'BSDMC'
methLevels(object)

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

Arguments

object	A BSDMC-class object
value	An integer matrix

Value

A matrix

A BSDMC-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)
methl <- methc / metht
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 <- cBSDMC(
  rowRanges = r1, methReads = methc, totalReads = metht,
  methLevels = methl, colData = cd1
)
methLevels(OBJ1)
methLevels(OBJ1) <- methl
```

`methReads-method` *methReads method*

Description

Returns `methReads` stored in `BSDMC-class`

Assigns `methReads` to `BSDMC-class`

Usage

```
methReads(object)

methReads(object) <- value

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

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

Arguments

`object`	A `BSDMC-class` object
`value`	An integer matrix

Value

A matrix

A `BSDMC-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)
methl <- methc / metht
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 <- cBSDMC(
  rowRanges = r1, methReads = methc, totalReads = metht,
  methLevels = methl, colData = cd1
)
methReads(OBJ1)
methReads(OBJ1) <- methc
```

`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 Bayesian functional regression models. The rows represent the CpG sites in `rowRanges` and the columns represent the samples in `colData`.
`rowRanges`	A `GRanges` or `GRangesList` object describing the ranges of interest. Names, if present, become the row names of the `SummarizedExperiment` object. The length of the `GRanges` or `GRangesList` must equal the number of rows of the matrices in `assays`. If `rowRanges` is missing, a `SummarizedExperiment` instance is returned.
`colData`	Object of class `'DataFrame'` containing information on variable values of the samples
`metadata`	A `list` of storing MCMC samples or DMCs
`object`	A `BSDMC-class` object

value	An integer matrix
name	A character list
obj1	A BSDMC-class
obj2	A BSDMC-class
files	A character list
file	A character
nCores	An integer value specifying the number of machine cores for parallel computing
mc.cores	An integer greater than 0
pSize	An integer value specifying the number of cytosines in a regrion to be used in a Bayesian functiona regression model for DMC detection
bwa	An integer value specifying the band-width size of B-spline basis matrix for a natural cubic spline for the group-specific effects of the Bayesian functional regression model
bwb	An integer value specifying the band-width size of B-spline basis matrix for a natural cubic spline for the individual-specific effects of the Bayesian functional regression model
nBurn	An integer value specifying the number of burn-in samples
nThin	An integer value specifying the thining number in MCMC
nMC	An integer value specifying the number of MCMC samples after burn-in
sdv	An double value specifying the standard deviation of priors
alpha	A numeric value specifying the level of α in credible interval $(1-\alpha)\%$
col	A character vector indicating which colors to alternate.
sfiles	A logical value indicating whether files to be saved or not.
region	An integer vector of length two specifying which subset of the object to be plotted
nSplit	A integer value specifying the number of subsets must be done for plotting the results of DMC identification
parList	A list specifying plots parameters, see par
...	other possible parameters

Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

plotDMCFB-method *plotDMCFB method*

Description

Plotting the results of DMC identifation stored in a BSDMC-class object

Usage

```
plotDMCFB(object, region, nSplit, parList)

## S4 method for signature 'BSDMC'
plotDMCFB(object, region, nSplit, parList)
```

Arguments

object	A BSDMC-class object
region	An integer vector of length two specifying which subset of the object to be plotted
nSplit	A integer value specifying the number of subsets must be done for plotting the results of DMC identification
parList	A list specifying plots parameters, see par

Value

Plot

Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

Examples

```
set.seed(1980)
nr <- 1000
nc <- 4
metht <- matrix(as.integer(runif(nr * nc, 0, 100)), nr)
methc <- matrix(rbinom(n = nr * nc, c(metht), prob = runif(nr * nc)), nr, nc)
methl <- methc / metht
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 <- cBSDMC(
  rowRanges = r1, methReads = methc, totalReads = metht,
  methLevels = methl, colData = cd1
```

```
)
OBJ2 <- findDMCFB(OBJ1,
  bwa = 10, bwb = 10, nBurn = 50, nMC = 50, nThin = 1,
  alpha = 0.05, nCores = 2, pSize = 500, sfiles = FALSE
)
plotDMCFB(OBJ2)
```

`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 `BSDMC-class` object

Author(s)

Farhad Shokoohi <shokoohi@icloud.com>

Examples

```
fn <- list.files(system.file("extdata", package = "DMCFB"))
fn.f <- list.files(system.file("extdata", package = "DMCFB"),
  full.names = TRUE
)
OBJ <- readBismark(fn.f, fn, mc.cores=1)
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` *totalReads method*

Description

Returns `totalReads` stored in `BSDMC-class`

Assigns `totalReads` to `BSDMC-class`

Usage

```
totalReads(object)

totalReads(object) <- value

## S4 method for signature 'BSDMC'
totalReads(object)

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

Arguments

`object`	A `BSDMC-class` object
`value`	An integer matrix

Value

A matrix

A `BSDMC-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)
methl <- methc / metht
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 <- cBSDMC(
  rowRanges = r1, methReads = methc, totalReads = metht,
  methLevels = methl, colData = cd1
)
totalReads(OBJ1)
totalReads(OBJ1) <- metht
```

Index

∗ **object**
BSDMC-class, 3

BSDMC *(*BSDMC-class*)*, 3
BSDMC-class, 3

cBSDMC, *2*
cBSDMC *(*cBSDMC-method*)*, 4
cBSDMC,matrix,matrix,matrix,GRanges-method *(*cBSDMC-method*)*, 4
cBSDMC-method, 4
combine *(*combine-method*)*, 6
combine,BSDMC,BSDMC-method *(*combine-method*)*, 6
combine-method, 6

DMCFB *(*DMCFB-package*)*, 2
DMCFB-package, 2

findDMCFB, *2*
findDMCFB *(*findDMCFB-method*)*, 7
findDMCFB,BSDMC-method *(*findDMCFB-method*)*, 7
findDMCFB-method, 7

GRanges, *5*, *11*
GRangesList, *5*, *11*

methLevels *(*methLevels-method*)*, 9
methLevels,BSDMC-method *(*methLevels-method*)*, 9
methLevels-method, 9
methLevels<- *(*methLevels-method*)*, 9
methLevels<-,BSDMC,matrix-method *(*methLevels-method*)*, 9
methReads *(*methReads-method*)*, 10
methReads,BSDMC-method *(*methReads-method*)*, 10
methReads-method, 10
methReads<- *(*methReads-method*)*, 10
methReads<-,BSDMC,matrix-method *(*methReads-method*)*, 10

par, *12*, *13*
params, 11
plotDMCFB, *2*
plotDMCFB *(*plotDMCFB-method*)*, 13
plotDMCFB,BSDMC-method *(*plotDMCFB-method*)*, 13
plotDMCFB-method, 13

readBismark, *2*
readBismark *(*readBismark-method*)*, 14
readBismark,character,character,numeric-method *(*readBismark-method*)*, 14
readBismark,character,data.frame,numeric-method *(*readBismark-method*)*, 14
readBismark,character,DataFrame,numeric-method *(*readBismark-method*)*, 14
readBismark-method, 14

SummarizedExperiment, *5*, *11*

totalReads *(*totalReads-method*)*, 15
totalReads,BSDMC-method *(*totalReads-method*)*, 15
totalReads-method, 15
totalReads<- *(*totalReads-method*)*, 15
totalReads<-,BSDMC,matrix-method *(*totalReads-method*)*, 15