Package ‘DMRScan’

October 17, 2017

Title Detection of Differentially Methylated Regions

Version 1.0.0

Description This package detects significant differentially methylated regions (for both qualitative and quantitative traits), using a scan statistic with underlying Poisson heuristics. The scan statistic will depend on a sequence of window sizes (# of CpGs within each window) and on a threshold for each window size. This threshold can be calculated by three different means: i) analytically using Siegmund et.al (2012) solution (preferred), ii) an important sampling as suggested by Zhang (2008), and a iii) full MCMC modeling of the data, choosing between a number of different options for modeling the dependency between each CpG.

biocViews Software, Technology, Sequencing, WholeGenome

Depends R (>= 3.4.0)

Imports Matrix, MASS, RcppRoll, ggplot2, methods, mvtnorm, stats, parallel

License GPL-3

LazyData true

Author Christian M Page [aut, cre], Linda Vos [aut], Trine B Rounge [ctb, dtc], Hanne F Harbo [ths], Bettina K Andreassen [aut]

Maintainer Christian M Page <`page.ntnu@gmail.com`>

RoxygenNote 5.0.1

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

DMRscan 2
DMRScan.methylationData 3
DMRScan.phenotypes 4
DMRScan_package 4
estimateThreshold 5
getRegions 6

head,RegionList-method . . . 7
length,Region-method . . . 7
makeCpGgenes . . . 8
makeCpGregions . . . 9
manyWindowSizeScanner . . . 10
names,Region-method . . . 10
nCpG . . . 11
oneWindowSizeScanner . . . 12
plot.Region . . . 12
pos . . . 13
print,Region-method . . . 14
pVal . . . 14
range,Region-method . . . 15
Region . . . 15
Region-class . . . 16
RegionList . . . 16
RegionList-class . . . 17
setRegion . . . 17
show,Region-method . . . 18
sort,RegionList-method . . . 18
tVal . . . 19
[. . . 19
[[. . . 20

Index **21**

`DMRscan` *DMR Scan function*

Description

DMR Scan function

Usage

```
DMRScan(observations, windowSize, windowThreshold = NULL, ...)
```

Arguments

`observations`	An object of type RegionList
`windowSize`	A sequence of windowSizes for the slidingWindow, must be an integer
`windowThreshold`	Optional argument with corresponding cut-off for each window. Will be estimated if not supplied.
`...`	Optional arguments to be pased to estimate_windowThreshold(), if no grid is specified.

Value

An object of type RegionList with signficantly differentially

Examples

```
## nProbeoad methylation data from chromosome 22
data(DMRScan.methylationData)
## nProbeoad phenotype (end-point for methylation data)
data(DMRScan.phenotypes)

## Test for an association between phenotype and Methylation
test.statistics <- apply(DMRScan.methylationData,1,function(x,y)
  summary(glm(y ~ x, family = binomial(link = "logit")))$coefficients[2,3],
                                                   y = DMRScan.phenotypes)
## Set chromosomal position to each test-statistic
positions <- data.frame(matrix(as.integer(unlist(strsplit(names(test.statistics), split="chr|[.]"))), ncol =
## Set clustering features
min.cpg <- 4  ## Minimum number of CpGs in a tested cluster
## Maxium distance (in base-pairs) within a cluster
## before it is broken up into two seperate cluster
max.gap <- 750

## Identify all clusters, and generate a list for each cluster
regions <- makeCpGregions(observations = test.statistics,
                          chr = positions[,1], pos = positions[,2],
                          maxGap = max.gap, minCpG = min.cpg)
## Number of CpGs in the slidingWindows, can be either a single number
## or a sequence of windowSizes
windowSizes <- 3:7
nCpG        <- nCpG(regions) ## Number of CpGs to be tested

# Estimate the windowThreshold, based on the number of CpGs and windowSizes
windowThresholds <- estimateWindowThreshold(nProbe = nCpG,
               windowSize = windowSizes, method = "sampling", mcmc = 10000)
## Run the slidingWindow
DMRScanResults   <- DMRScan(observations = regions,
                            windowSize = windowSizes,
                            windowThreshold = windowThresholds)
## Print the result
print(DMRScanResults)
```

`DMRScan.methylationData` *DMRScan*

Description

Bi-sulfite sequencing data of known CpGs at chromosome 22 from 100 Finish teens, sampled from the two extreme BMI quantiles. See "Genome-wide DNA methylation in saliva and body size of adolescent girls", TB Rounge, CM Page, M Lepisto, E Pekka, and BK Andreassen and E Weiderpass, Epigenomics 8.11 (2016): 1495-1505.

Examples

```
data(DMRScan.methylationData)
head(DMRScan.methylationData)
```

`DMRScan.phenotypes` *DMRScan*

Description

Phenotypes for methylation data, indicating case control status. See "Genome-wide DNA methylation in saliva and body size of adolescent girls", TB Rounge, CM Page, M Lepisto, E Pekka, and BK Andreassen and E Weiderpass, Epigenomics 8.11 (2016): 1495-1505.

Examples

```
data(DMRScan.phenotypes)
table(DMRScan.phenotypes)
```

`DMRScan_package` *DMRScan: An R-package for identification of Differentially Metylated Regions*

Description

DMRScan: An R-package for identification of Differentially Metylated Regions

Arguments

`observations`	An object of type RegionList
`windowSize`	A sequence of windowSizes for the slidingWindow, must be an integer
`windowThreshold`	Optional argument with corresponding cut-off for each window. Will be estimated if not supplied.
`...`	Optional arguments to be pased to estimate_windowThreshold(), if no grid is specified.

Value

An object of type RegionList with signficantly differentially

Author(s)

Christian Page, <`page.ntnu@gmail.com`>

References

`http://Some_link_to_BMC-bioInfomatics.com`

Examples

```
## nProbeoad methylation data from chromosome 22
data(DMRScan.methylationData)
## nProbeoad phenotype (end-point for methylation data)
data(DMRScan.phenotypes)

## Test for an association between phenotype and Methylation
test.statistics <- apply(DMRScan.methylationData,1,function(x,y)
  summary(glm(y ~ x, family = binomial(link = "logit")))$coefficients[2,3],
                                                    y = DMRScan.phenotypes)
## Set chromosomal position to each test-statistic
positions <- data.frame(matrix(as.integer(unlist(strsplit(names(test.statistics), split="chr|[.]"))), ncol =
## Set clustering features
min.cpg <- 4  ## Minimum number of CpGs in a tested cluster
## Maxium distance (in base-pairs) within a cluster
## before it is broken up into two seperate cluster
max.gap <- 750

## Identify all clusters, and generate a list for each cluster
regions <- makeCpGregions(observations = test.statistics,
                          chr = positions[,1], pos = positions[,2],
                          maxGap = max.gap, minCpG = min.cpg)
## Number of CpGs in the slidingWindows, can be either a single number
## or a sequence of windowSizes
windowSizes <- 3:7
nCpG        <- nCpG(regions) ## Number of CpGs to be tested

# Estimate the windowThreshold, based on the number of CpGs and windowSizes
windowThresholds <- estimateWindowThreshold(nProbe = nCpG,
               windowSize = windowSizes, method = "sampling", mcmc = 10000)
## Run the slidingWindow
DMRScanResults   <- DMRScan(observations = regions,
                            windowSize = windowSizes,
                            windowThreshold = windowThresholds)
## Print the result
print(DMRScanResults)
```

`estimateThreshold` *Estimate window thresholds*

Description

Estimate window thresholds for sliding window, one unique value for each window size

Usage

```
estimateWindowThreshold(nProbe, windowSize, method = "siegmund",
  mcmc = 1000, nCPU = 1, submethod = "ar", ...)
```

Arguments

`nProbe`	The number of probes (CpGs) in the study.

windowSize	The different window sizes to be tested. Must be either one, or an ordered sequence of integers.
method	Gives the method by which the threshold is calculated. Can be either an analytical solution "siegmund", provided by Siegnumd et.al (2012), or an iterative process; either importance sampling "sampling", as suggested by Zhang (2012) or a full MCMC model "mcmc" which can account for any dependency structure, wich is pass to arima.sim, with ...
mcmc	The number of MCMC iterations to be used, when using either Important Sampling ("zhang") or MCMC estimation of the threshold.
nCPU	When calculating the thresholds on a cluster, how many CPUs should be used. This option is only compatible with the 'mcmc' method.
submethod	A character string indicating if an AR(5) or ARIMA model should be used. In the AR(5), the index runs from -2 to 2. A regular AR(p) model can be obtaine using ARIMA(p,0,0) instead.
...	Optinal parameters pased on to arima(), when simulating data using the mcmc option, see arima.sim()

Value

Returns a vector of the threshold for each window size

Examples

```
thresholdGrid <- estimateWindowThreshold(nProbe = 1000,
                                         windowSize = 3:8, method = "siegmund")
```

getRegions *Method getRegions*

Description

Method getRegions

getRegions for Region List

Usage

```
getRegions(x)
```

Arguments

x	An object of type RegionList

Value

An object of type Region

A region from a RegionList

Examples

```
someEmptyRegions <- RegionList(3L)
# To get back three empty regions
getRegions(someEmptyRegions)
```

`head,RegionList-method` *Cat the head of a list of regions in a RegionList object*

Description

Cat the head of a list of regions in a RegionList object

Usage

```
## S4 method for signature 'RegionList'
head(x, n = 10L)
```

Arguments

`x`	An object to be printed of type RegionList
`n`	The number of regions to be printed when the RegionList is longer than n

Value

The top regins in a RegionList

`length,Region-method` *Calculate the length of a region in terms of CpGs*

Description

Calculate the length of a region in terms of CpGs

Get the number of regions in a RegionList

Usage

```
## S4 method for signature 'Region'
length(x)

## S4 method for signature 'RegionList'
length(x)
```

Arguments

`x`	A RegionList object

Value

The number of CpGs in a Region

The number of CpGs in a RegionList

`makeCpGgenes` *Cluster*

Description

Clustger CpGs together in genes based on annotation

Usage

```
makeCpGgenes(observations, chr, pos, gene, minCpG = 2)
```

Arguments

`observations`	Vector of corresponding observed T-value for each CpG, must be ordered in the same way as chr and pos
`chr`	Vector of chromosome location for each CpG
`pos`	Vector giving base pair position for each CpG If unsorted, use order(chr,pos) to sort the genomic positions within each chromosome.
`gene`	A vector asigning each probe to a gene.
`minCpG`	Minimum number of CpGs allowed in each region to be considered. Default is set to at least 2 CpGs within each region.

Value

The suplied observations ordered into into a list, with one entry for each CpG region.

Examples

```
data(DMRScan.methylationData) ## Load methylation data from chromosome 22
data(DMRScan.phenotypes) ## Load phenotype (end-point for methylation data)

## Test for an association between phenotype and Methylation
testStatistics <- apply(DMRScan.methylationData,1,function(x,y)
 summary(glm(y ~ x, family = binomial(link = "logit")))$coefficients[2,3],
  y = DMRScan.phenotypes)

## Set chromosomal position to each test-statistic
pos <- data.frame(matrix(as.integer(unlist(strsplit(names(testStatistics),
 split="chr|[.]"))), ncol = 3, byrow = TRUE))[,-1]

## Set clustering features
minCpG   <- 3  ## Minimum number of CpGs in a tested cluster
gene     <- sample(paste("Gene",1:100,sep=""),
                           length(testStatistics),replace=TRUE)
regions  <- makeCpGgenes(observations = testStatistics,
                         chr = pos[,1], pos = pos[,2],
                         gene = gene, minCpG = minCpG)
```

`makeCpGregions` *Cluster*

Description

Clustger CpGs together in regions based on proximity

Usage

```
makeCpGregions(observations, chr, pos, maxGap = 500, minCpG = 2)
```

Arguments

`observations`	Vector of corresponding observed T-value for each CpG, must be ordered in the same way as chr and pos
`chr`	Vector of chromosome location for each CpG
`pos`	Vector giving base pair position for each CpG If unsorted, use order(chr,pos) to sort the genomic positions within each chromosome.
`maxGap`	Maximum allowed base pair gap within a cluster. Default is set to 500.
`minCpG`	Minimum number of CpGs allowed in each region to be considered. Default is set to at least 2 CpGs within each region.

Value

The suplied observations ordered into into a RegionList object. To be parsed further into DMRScan()

Examples

```
data(DMRScan.methylationData) ## Load methylation data from chromosome 22
data(DMRScan.phenotypes) ## Load phenotype (end-point for methylation data)

## Test for an association between phenotype and Methylation
testStatistics <- apply(DMRScan.methylationData,1,function(x,y)
 summary(glm(y ~ x, family = binomial(link = "logit")))$coefficients[2,3],
 y = DMRScan.phenotypes)

## Set chromosomal position to each test-statistic
pos<- data.frame(matrix(as.integer(unlist(strsplit(names(testStatistics),
split="chr|[.]"))), ncol = 3, byrow = TRUE))[,-1]

## Set clustering features
minCpG <- 3  ## Minimum number of CpGs in a tested cluster
## Maxium distance (in base-pairs) within a cluster before it is
## broken up into two seperate cluster
maxGap <- 750
regions <- makeCpGregions(observations = testStatistics, chr = pos[,1],
                            pos = pos[,2], maxGap = maxGap, minCpG = minCpG)
```

manyWindowSizeScanner *Method Fixed window size scan for a sequence of window sizes*

Description

Method Fixed window size scan for a sequence of window sizes

Usage

```
manyWindowSizeScanner(region, windowThreshold, windowSize)

## S4 method for signature 'RegionList'
manyWindowSizeScanner(region, windowThreshold,
  windowSize)

## S4 method for signature 'Region'
manyWindowSizeScanner(region, windowThreshold, windowSize)
```

Arguments

`region`	Object of type Region or RegionList
`windowThreshold`	Vector of window thresholds
`windowSize`	Vector of window sizes to be tested on regions

Value

A list of which windows that are significant

Examples

```
## Not run
```

names,Region-method *Get the names of all probes within a region*

Description

Get the names of all probes within a region

Get the names of all probes in a study

Usage

```
## S4 method for signature 'Region'
names(x)

## S4 method for signature 'RegionList'
names(x)
```

Arguments

`x`	An object of type Region

Value

The names of individual CpGs in a Region

A character vector of all CpG ids in a RegionList

`nCpG` *Method nCpG*

Description

Method nCpG

Get the number of CpGs i a region

Get the number of CpGs in a RegionList

Usage

```
nCpG(x)

## S4 method for signature 'Region'
nCpG(x)

## S4 method for signature 'RegionList'
nCpG(x)
```

Arguments

`x`	An opbject of type Region or RegionList

Value

The number of CpGs in an object

Examples

```
someEmptyRegions <- RegionList(3L)
# The number of CpGs in this regions is 0
nCpG(someEmptyRegions)
```

oneWindowSizeScanner *Method Fixed window size scan for one window size*

Description

Method Fixed window size scan for one window size

Usage

```
oneWindowSizeScanner(region, windowThreshold, windowSize)

## S4 method for signature 'RegionList'
oneWindowSizeScanner(region, windowThreshold, windowSize)

## S4 method for signature 'Region'
oneWindowSizeScanner(region, windowThreshold, windowSize)
```

Arguments

`region`	Object of type Region or RegionList
`windowThreshold`	Vector of window thresholds
`windowSize`	Vector of window sizes to be tested on regions

Value

A list of which windows that are significant

Examples

```
## Not run
```

plot.Region *Plot DMRs of type Region*

Description

Plot DMRs of type Region

Usage

```
## S3 method for class 'Region'
plot(x, ...)
```

Arguments

`x`	A Region object to be ploted. Can be subsetted from RegionList
`...`	Inherited from plot()

Value

A plot object

`pos` *Method pos*

Description

Method pos

Get the chromosomal coordinates for a Region

Get the chromosomal coordinates for a list of regions in a RegionList object

Usage

```
pos(region)

## S4 method for signature 'Region'
pos(region)

## S4 method for signature 'RegionList'
pos(region)
```

Arguments

`region`	An opbject of type Region or RegionList

Value

An integer vector of positions for each probe site

Examples

```
#Number of probes is n = 10
nCpG <- 10
region <- Region(tValues    = rnorm(nCpG),
                 position   = 1:nCpG,
                 chromosome = "3")
## Genomic coordinates for Region
pos(region)
```

`print,Region-method` *Print a region*

Description

Print a region

Print a number of regions in a RegionList

Usage

```
## S4 method for signature 'Region'
print(x, ...)

## S4 method for signature 'RegionList'
print(x)
```

Arguments

`x`	Object of type Region
`...`	Has no function

Value

An print object of a Region class

A printed object of all regions in a RegionList

`pVal` *Method get pvalue*

Description

Method get pvalue

Get p-values for a region

Get p-values for a list of regions (RegionList)

Usage

```
pVal(region, n = 12)

## S4 method for signature 'Region'
pVal(region, n = 12)

## S4 method for signature 'RegionList'
pVal(region, n = 12)
```

Arguments

`region`	An object of type Region or RegionList
`n`	The number of digits to be presented. Default is 10

Value

A numeric vector of p-values

Examples

```
#Number of probes is n = 10
nCpG <- 10
region <- Region(tValues    = rnorm(nCpG),
                 position   = 1:nCpG,
                 chromosome = "3",
                 pVal       = runif(1))
## Pvalues for Region
pVal(region)
```

`range,Region-method` *Get the genomic position of a Region*

Description

Get the genomic position of a Region

Usage

```
## S4 method for signature 'Region'
range(x)
```

Arguments

`x`	An object of type Region

Value

A character giving the genomic position

`Region` *Shorthand for initializing region*

Description

Shorthand for initializing region

Usage

```
Region(tValues, position, chromosome, pVal, id)
```

Arguments

`tValues`	A vector of test statistics
`position`	A vector of position for each test statistc
`chromosome`	An character describing the chromosome (1-22, X,Y)
`pVal`	The P value of a region, set to numeric() if not given.
`id`	The names of each probe in the region

Value

An object of type Region

An object of type Region

Examples

```
#Number of probes is n = 10
nCpG <- 10
region <- Region(tValues    = rnorm(nCpG),
                 position   = 1:nCpG,
                 chromosome = "3",
                 id         = paste("CpG",1:nCpG,sep="_"),
                 pVal       = runif(1))
```

Region-class *Object of type Region*

Description

Class Region is a collection of test statistics for a set of CpGs within a short genomic range

---

RegionList *Shorthand for initializing RegionList*

Description

Shorthand for initializing RegionList

Usage

```
RegionList(nRegions, regions)
```

Arguments

nRegions	The number of regions to be placed
regions	The regions to be included

Value

An object of type RegionList

Examples

```
# An empty list of 3 regions
RegionList(3L)
```

RegionList-class *Class RegionList Class* RegionList *is a collection of Regions*

Description

Class RegionList

Class RegionList is a collection of Regions

setRegion *Method setRegion*

Description

Method setRegion

Update a RegionList object

Usage

```
setRegion(x, i, ...)

## S4 method for signature 'RegionList'
setRegion(x, i, region)
```

Arguments

x	A region
i	an index
...	To be pased to Region()
region	An object of type Region to be inseted in RegionList

Value

An updated version of RegionList x, with a new Region at index i

Examples

```
## A region list with 3 regions
regList <- RegionList(3L)
#Number of probes in first is n = 10
nCpG <- 10
region <- Region(tValues    = rnorm(nCpG),
                 position   = 1:nCpG,
                 chromosome = "3")
## Set first region in regList to region
regList <- setRegion(regList,i = 1, region)
```

`show,Region-method` *Show a region*

Description

Show a region

Usage

```
## S4 method for signature 'Region'
show(object)
```

Arguments

`object`	The region to be desplied, of type Region

Value

Cat a region to screen

`sort,RegionList-method` *Sort a set of regions on p-value in a RegionList object*

Description

Sort a set of regions on p-value in a RegionList object

Usage

```
## S4 method for signature 'RegionList'
sort(x, decreasing = FALSE)
```

Arguments

`x`	An object of type RegionList
`decreasing`	Inherited from base

Value

An updated RegionList, sorted on empirical p-values

tVal — *Method get T statistic for a region*

Description

Method get T statistic for a region

Get test statistic for an object of type Region

Get test statistic for all regins within a RegionList class

Usage

```
tVal(region, ...)

## S4 method for signature 'Region'
tVal(region, index = NULL)

## S4 method for signature 'RegionList'
tVal(region, index = NULL)
```

Arguments

`region`	An opbject of type Region or RegionList
`...`	Index
`index`	Index to extract

Value

A numeric vector of t-values for a Region or RegionList

Examples

```
#Number of probes is n = 10
nCpG <- 10
region <- Region(tValues    = rnorm(nCpG),
                 position   = 1:nCpG,
                 chromosome = "3")
## T values for Region
tVal(region)
```

[— *Get Object Region*

Description

Get Object Region

Arguments

`x`	An object of type RegionList
`i`	Index, which region to extract
`j`	(Not used)
`...`	(not used)
`drop`	If drop is used

Value

A region from a RegionList of class "list"

`[[` *Get Object Region*

Description

Get Object Region

Usage

```
## S4 method for signature 'RegionList'
x[[i, j, ..., drop]]
```

Arguments

`x`	An object of type RegionList
`i`	Index, which region to extract
`j`	(Not used)
`...`	(not used)
`drop`	If drop is used

Value

A region from a RegionList with class "Region"

Index

∗Topic **CpG**
    makeCpGgenes, 8
    makeCpGregions, 9
∗Topic **DMR,**
    DMRScan_package, 4
∗Topic **DMRScan**
    DMRscan, 2
    DMRScan_package, 4
∗Topic **Regions**
    makeCpGgenes, 8
    makeCpGregions, 9
∗Topic **datasets**
    DMRScan.methylationData, 3
∗Topic **dataset**
    DMRScan.phenotypes, 4
[, 19
[,RegionList,ANY,ANY,ANY-method *([)*, 19
[[, 20
[[,RegionList-method *([[)*, 20

DMRScan *(DMRscan)*, 2
DMRscan, 2
DMRScan.methylationData, 3
DMRScan.phenotypes, 4
DMRScan_package, 4
DMRScan_package-package
        (DMRScan_package), 4

estimateThreshold, 5
estimateWindowThreshold
        (estimateThreshold), 5

getRegions, 6
getRegions,RegionList-method
        (getRegions), 6

head,RegionList-method, 7

length,Region-method, 7
length,RegionList-method
        (length,Region-method), 7

makeCpGgenes, 8
makeCpGregions, 9
manyWindowSizeScanner, 10
manyWindowSizeScanner,Region-method
        (manyWindowSizeScanner), 10
manyWindowSizeScanner,RegionList-method
        (manyWindowSizeScanner), 10

names,Region-method, 10
names,RegionList-method
        (names,Region-method), 10
nCpG, 11
nCpG,Region-method *(nCpG)*, 11
nCpG,RegionList-method *(nCpG)*, 11

oneWindowSizeScanner, 12
oneWindowSizeScanner,Region-method
        (oneWindowSizeScanner), 12
oneWindowSizeScanner,RegionList-method
        (oneWindowSizeScanner), 12

plot *(plot.Region)*, 12
plot.Region, 12
pos, 13
pos,Region-method *(pos)*, 13
pos,RegionList-method *(pos)*, 13
print,Region-method, 14
print,RegionList-method
        (print,Region-method), 14
pVal, 14
pVal,Region-method *(pVal)*, 14
pVal,RegionList-method *(pVal)*, 14

range,Region-method, 15
Region, 15
Region-class, 16
RegionList, 16
RegionList-class, 17
Rt *(oneWindowSizeScanner)*, 12
Rt,Region-method
        (oneWindowSizeScanner), 12
Rt,RegionList-method
        (oneWindowSizeScanner), 12

setRegion, 17
setRegion,RegionList-method
        (setRegion), 17
show,Region-method, 18

sort,RegionList-method, 18
St *(*manyWindowSizeScanner*)*, 10

tVal, 19
tVal,Region-method *(*tVal*)*, 19
tVal,RegionList-method *(*tVal*)*, 19