

# Package ‘MethylSeekR’

October 17, 2020

**Type** Package

**Title** Segmentation of Bis-seq data

**Version** 1.28.0

**Date** 2014-7-1

**Author** Lukas Burger, Dimos Gaidatzis, Dirk Schubeler and Michael Stadler

**Maintainer** Lukas Burger <Lukas.Burger@fmi.ch>

**Description** This is a package for the discovery of regulatory regions from Bis-seq data

**Depends** rtracklayer (>= 1.16.3), parallel (>= 2.15.1), mhsmm (>= 0.4.4)

**Imports** IRanges (>= 1.16.3), BSgenome (>= 1.26.1), GenomicRanges (>= 1.10.5), geneplotter (>= 1.34.0), graphics (>= 2.15.2), grDevices (>= 2.15.2), parallel (>= 2.15.2), stats (>= 2.15.2), utils (>= 2.15.2)

**Suggests** BSgenome.Hsapiens.UCSC.hg18

**License** GPL (>=2)

**biocViews** Sequencing, MethylSeq, DNAMethylation

**git\_url** <https://git.bioconductor.org/packages/MethylSeekR>

**git\_branch** RELEASE\_3\_11

**git\_last\_commit** 202741a

**git\_last\_commit\_date** 2020-04-27

**Date/Publication** 2020-10-16

## R topics documented:

MethylSeekR-package . . . . .	2
calculateFDRs . . . . .	3
plotAlphaDistributionOneChr . . . . .	5
plotFinalSegmentation . . . . .	6
plotPMDSegmentation . . . . .	7
readMethylome . . . . .	9
readSNPTable . . . . .	10
removeSNPs . . . . .	11

savePMDSegments . . . . .	12
saveUMRLMRSegments . . . . .	13
segmentPMDs . . . . .	14
segmentUMRsLMRs . . . . .	15

<b>Index</b>	<b>17</b>
--------------	-----------

---

MethylSeekR-package    *Segmentation of Bis-seq methylation data*

---

## Description

This is a package for the discovery of regulatory regions from Bis-seq data

## Details

Package: MethylSeekR  
 Type: Package  
 Version: 1.0  
 Date: 2012-10-10  
 License: None

## Author(s)

Lukas Burger

Maintainer: Lukas Burger <Lukas.Burger@fmi.ch>

## References

Stadler, Murr, Burger et al, DNA-binding factors shape the mouse methylome at distal regulatory regions, Nature 2011.

## Examples

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqlengths(Hsapiens)

# read methylation data
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
  package="MethylSeekR")
meth.gr <- readMethylome(FileName=methFname, seqLengths=sLengths)

#read SNP data
snpFname <- system.file("extdata", "SNVs_hg18_chr22.tab",
  package="MethylSeekR")
```

```

snps.gr <- readSNPTable(FileName=snpFname, seqLengths=sLengths)

# remove SNPs
meth.gr <- removeSNPs(meth.gr, snps.gr)

#calculate alpha distribution for one chromosome
plotAlphaDistributionOneChr(m=meth.gr, chr.sel="chr22", num.cores=1)

#segment PMDs
PMDsegments.gr <- segmentPMDs(m=meth.gr, chr.sel="chr22", num.cores=1,
seqLengths=sLengths)

#plot PMD segmentation examples
plotPMDSegmentation(m=meth.gr, segs=PMDsegments.gr, numRegions=1)

#save PMD segments
savePMDSegments(PMDs=PMDsegments.gr, GRangesFilename="PMDs.gr.rds",
  TableFilename="PMDs.tab")

#load CpG islands
library(rtracklayer)
session <- browserSession()
genome(session) <- "hg18"
query <- ucscTableQuery(session, "cpgIslandExt")
CpGislands.gr <- track(query)
genome(CpGislands.gr) <- NA
CpGislands.gr <- resize(CpGislands.gr, 5000, fix="center")

#calculate FDRs
stats <- calculateFDRs(m=meth.gr, CGIs=CpGislands.gr,
PMDs=PMDsegments.gr, num.cores=1)

# select FDR cut-off and determine segmentation parameters
FDR.cutoff <- 5
m.sel <- 0.5
n.sel=as.integer(names(stats$FDRs[as.character(m.sel), ]
[stats$FDRs[as.character(m.sel), ]<FDR.cutoff])[1])

#segment UMRs and LMRs
UMRLMRsegments.gr <- segmentUMRSLMRs(m=meth.gr, meth.cutoff=m.sel,
nCpG.cutoff=n.sel, PMDs=PMDsegments.gr, num.cores=1,
myGenomeSeq=Hsapiens, seqLengths=sLengths)

#plot final segmentation including PMDs, UMRs and LMRs
plotFinalSegmentation(m=meth.gr, segs=UMRLMRsegments.gr, PMDs=PMDsegments.gr,
meth.cutoff=m.sel, numRegions=1)

#save UMRs and LMRs
saveUMRLMRSegments(segs=UMRLMRsegments.gr, GRangesFilename="UMRSLMRs.gr.rds",
  TableFilename="UMRSLMRs.tab")

```

**Description**

This function calculates the false discovery rate (FDR) based on randomized Bis-seq data.

**Usage**

```
calculateFDRs(m, CGIs, PMDs = NA, pdfFilename=NULL, num.cores = 1,
nCpG.smoothing = 3, meth.cutoffs = seq(0.3, 0.7, by=0.1), nCpG.cutoffs =
seq(1, 6, by=1), minCover = 5)
```

**Arguments**

<code>m</code>	GRanges object containing the methylation data.
<code>CGIs</code>	A GRanges object of CpG island coordinates. All CpGs overlapping CpG islands will be removed for the randomization.
<code>PMDs</code>	The GRanges object of the PMDs. Set to either the return value of the function <code>segmentPMDs</code> (see example) or to <code>NA</code> (default) if there are no PMDs.
<code>pdfFilename</code>	Name of the pdf file in which the figure is saved. If no name is provided (default), the figure is printed to the screen.
<code>num.cores</code>	Number of cores used for the calculations.
<code>nCpG.smoothing</code>	The number of consecutive CpGs that the methylation levels are averaged over.
<code>meth.cutoffs</code>	A vector containing the cut-offs in methylation for which the FDR should be calculated. Numbers must be between 0 and 1.
<code>nCpG.cutoffs</code>	A vector containing the cut-offs on the minimal number of CpGs per region for which the FDR should be calculated.
<code>minCover</code>	Only CpGs with a coverage of at least <code>minCover</code> reads will be used.

**Value**

A list containing a matrix with FDR values and a matrix with the number of inferred segments for each methylation cut-off (rows) and each cut-off on the minimal number of CpGs per region (columns). The function creates a figure showing the relationship between the methylation cut-off, the cut-off on the minimal number of CpGs per region, the number of inferred segments and the FDR. The figure is either printed to the screen (default) or saved as a pdf if a filename is provided.

**Author(s)**

Lukas Burger [lukas.burger@fmi.ch](mailto:lukas.burger@fmi.ch)

**Examples**

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqlengths(Hsapiens)

# read methylation data
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
package="MethylSeekR")
meth.gr <- readMethylome(FileName=methFname, seqLengths=sLengths)
```

```
#load CpG islands
library(rtracklayer)
session <- browserSession()
genome(session) <- "hg18"
query <- ucscTableQuery(session, "cpgIslandExt")
CpGislands.gr <- track(query)
genome(CpGislands.gr) <- NA
CpGislands.gr <- resize(CpGislands.gr, 5000, fix="center")

#calculate FDRs, assuming no PMDs
stats <- calculateFDRs(m=meth.gr, CGIs=CpGislands.gr)
```

---

plotAlphaDistributionOneChr

*Calculate and plot alpha distribution.*

---

## Description

This function calculates the alpha values for a selected chromosome and plots the distribution of alpha values. The shape of the distribution is indicative of the presence or absence of partially methylated domains (PMDs).

## Usage

```
plotAlphaDistributionOneChr(m, chr.sel, pdfFilename = NULL, num.cores = 1, nCGbin = 101)
```

## Arguments

<code>m</code>	Methylation GRanges object.
<code>chr.sel</code>	Selected chromosome for which alpha values are calculated. Must be one of the sequence levels of <code>m</code> .
<code>pdfFilename</code>	Name of the pdf file in which the figure is saved. If no name is provided (default), the figure is printed to the screen.
<code>num.cores</code>	The number of cores that are used for the calculation (default 1).
<code>nCGbin</code>	The number of CpGs in each sliding window used to calculate alpha (default 101). The default is highly recommended.

## Value

No return value. The function creates a figure showing the alpha distribution for the selected chromosome that is either printed to the screen (default) or saved as a pdf if a filename is provided.

## Author(s)

Lukas Burger [lukas.burger@fmi.ch](mailto:lukas.burger@fmi.ch)

**Examples**

```

library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqlengths(Hsapiens)

# read methylation data
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
package="MethylSeekR")
meth.gr <- readMethylome(FileName=methFname, seqLengths=sLengths)

#calculate alpha distribution for one chromosome
plotAlphaDistributionOneChr(m=meth.gr, chr.sel="chr22", num.cores=1)

```

---

plotFinalSegmentation *Plotting final segmentation*

---

**Description**

This function plots the final segmentations, including PMDs, UMRs and LMRs.

**Usage**

```
plotFinalSegmentation(m, segs, PMDs = NA, meth.cutoff, numRegions = 1,
pdfFilename = NULL, minCover = 5, nCpG.smoothing = 3)
```

**Arguments**

<code>m</code>	GRanges object containing the methylation data.
<code>segs</code>	GRanges object containing the UMR/LMR segmentation. Return value of the <code>segmentUMRsLMRs</code> function (see example).
<code>PMDs</code>	GRanges object of PMDs. Set to either the return value of the function <code>segmentPMDs</code> (see example) or to NA if there are no PMDs.
<code>meth.cutoff</code>	Cut-off on methylation for calling hypomethylated regions.
<code>numRegions</code>	The number of randomly chosen regions to be plotted. The default (1) can only be changed if a <code>pdfFilename</code> is provided (see below).
<code>pdfFilename</code>	Name of the pdf file in which the figure is saved. If no name is provided (default), the figure is printed to the screen.
<code>minCover</code>	Only CpGs with a coverage of at least <code>minCover</code> reads will be used for plotting.
<code>nCpG.smoothing</code>	The number of consecutive CpGs that the methylation levels are averaged over.

**Value**

No return value. The function creates a figure showing the inferred segmentation for a randomly chosen region. The figure is either printed to the screen (default) or saved as a pdf if a filename is provided. If a filename (pdfFilename) is provided, several regions (set via the numRegions argument) can be plotted and saved in a multi-page pdf file. The randomly chosen region that is displayed is broken up into 3 pairs of panels, where in each pair the same region is shown twice, once with raw methylation levels (top) and once with methylation levels smoothed over 3 consecutive CpGs (bottom). In both cases only CpGs with a coverage of at least minCover reads are shown. The raw data best illustrates the disordered nature of methylation levels in PMDs, whereas the smoothed methylation levels more clearly show UMRs and LMRs. In all figures, UMRs are shown as blue squares (placed at the middle of the identified segment), LMRs as red triangles (placed at the middle of the identified segment) and PMDs as green bars (extending over the entire PMD). The cut-off on methylation (meth.cutoff) to determine UMRs and LMRs is shown as a grey dashed line.

**Author(s)**

Lukas Burger lukas.burger@fmi.ch

**Examples**

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqLengths(Hsapiens)

# read methylation data
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
package="MethylSeekR")
meth.gr <- readMethylome(FileName=methFname, seqLengths=sLengths)

FDR.cutoff <- 5
m.sel <- 0.5
n.sel <- 3

#segment UMRs and LMRs, assuming no PMDs
UMRLMRsegments.gr <- segmentUMRSLMRs(m=meth.gr, meth.cutoff=m.sel,
nCpG.cutoff=n.sel, myGenomeSeq=Hsapiens, seqLengths=sLengths)

#plot final segmentation, assuming no PMDs
plotFinalSegmentation(m=meth.gr, segs=UMRLMRsegments.gr, meth.cutoff=m.sel, numRegions=1)
```

---

plotPMDSegmentation *Plotting the PMD Segmentation*

---

**Description**

This function generates a figure showing the PMD segmentation in a randomly chosen region.

**Usage**

```
plotPMDSegmentation(m, segs, numRegions = 1, pdfFilename=NULL, minCover = 5)
```

**Arguments**

m	GRanges object containing the methylation data.
segs	GRanges object containing the PMD segmentation. Return value of the segmentPMDs function (see example).
numRegions	The number of randomly chosen regions to be plotted. The default (1) can only be changed if a pdfFilename is provided (see below).
pdfFilename	Name of the pdf file in which the figure is saved. If no name is provided (default), the figure is printed to the screen.
minCover	Only CpGs with a coverage of at least minCover reads will be used for plotting.

**Value**

No return value. The function creates a figure showing the inferred segmentation for a randomly chosen region. The figure is either printed to the screen (default) or saved as a pdf if a filename is provided. If a filename (pdfFilename) is provided, several regions (set via the numRegions argument) can be plotted and saved in a multi-page pdf file. The randomly chosen region that is displayed is broken up into 6 panels and in each panel, the raw (ie unsmoothed) methylation levels of all CpGs with a minimal coverage of 5 reads are shown. PMDs are indicated as green bars, extending over the entire PMD.

**Author(s)**

Lukas Burger [lukas.burger@fmi.ch](mailto:lukas.burger@fmi.ch)

**Examples**

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqlengths(Hsapiens)

# read methylation data
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
package="MethylSeekR")
meth.gr <- readMethylome(FileName=methFname, seqLengths=sLengths)

#segment PMDs
PMDsegments.gr <- segmentPMDs(m=meth.gr, chr.sel="chr22", seqLengths=sLengths)

#plot PMD segmentation examples
plotPMDSegmentation(m=meth.gr, segs=PMDsegments.gr, numRegions=1)
```



---

readMethylome	<i>Load Bis-seq data</i>
---------------	--------------------------

---

### Description

Loading Bis-seq data from tab-delimited file or saved GRanges object

### Usage

```
readMethylome(FileName, seqLengths, format = "text")
```

### Arguments

FileName	File name.
seqLengths	A named vector indicating the chromosome lengths of the genome used.
format	File format. If format is set to "text" (default), the argument FileName should refer to a tab-delimited text file in the format: chromosome position T M, where each line stands for a CpG, the position refers to the C of the CpG (on the plus strand), T is the total number of reads (total counts) covering the CpG and M the total number of reads without C to T conversion at the C of the CpG (methylation counts). If format="GRanges", the file is assumed to be a GRanges object, containing T and M as first and second data-value entries, saved in rds format.

### Value

A GRanges object containing the coordinates, total (T) and methylated counts (M)

### Author(s)

Lukas Burger [lukas.burger@fmi.ch](mailto:lukas.burger@fmi.ch)

### Examples

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqLengths(Hsapiens)

# read methylation data
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
package="MethylSeekR")
meth.gr <- readMethylome(FileName=methFname, seqLengths=sLengths)
```

readSNPTable                      *Load SNP table*

---

### Description

Loading SNPs from tab-delimited file or saved GRanges object

### Usage

```
readSNPTable(fileName, seqLengths, format = "text")
```

### Arguments

fileName	File Name.
seqLengths	A named vector indicating the chromosome lengths of the genome used.
format	File format. If format is set to "text", the argument fileName should refer to a tab-delimited text file in the format: chromosome position, where each line stands for a SNP. If format="GRanges", the file is assumed to be a GRanges object, containing the SNP coordinates, saved in rds format.

### Value

A GRanges object containing the coordinates of the SNPs.

### Author(s)

Lukas Burger [lukas.burger@fmi.ch](mailto:lukas.burger@fmi.ch)

### Examples

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqLengths(Hsapiens)

#read SNP data
snpFname <- system.file("extdata", "SNVs_hg18_chr22.tab",
package="MethylSeekR")
snps.gr <- readSNPTable(fileName=snpFname, seqLengths=sLengths)
```

---

removeSNPs	<i>Remove CpGs overlapping SNPs</i>
------------	-------------------------------------

---

**Description**

Removes CpGs that overlap with SNPs from methylation GRanges object.

**Usage**

```
removeSNPs(m, snps)
```

**Arguments**

m	GRanges object containing the methylation data.
snps	GRanges object containing the SNPs.

**Value**

The methylation GRanges object (m) with all CpGs overlapping SNPs removed.

**Author(s)**

Lukas Burger [lukas.burger@fmi.ch](mailto:lukas.burger@fmi.ch)

**Examples**

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqlengths(Hsapiens)

# read methylation data
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
package="MethylSeekR")
meth.gr <- readMethylome(FileName=methFname, seqLengths=sLengths)

#read SNP data
snpFname <- system.file("extdata", "SNVs_hg18_chr22.tab",
package="MethylSeekR")
snps.gr <- readSNPTable(FileName=snpFname, seqLengths=sLengths)

# remove SNPs
meth.gr <- removeSNPs(meth.gr, snps.gr)
```

---

savePMDSegments      *Save PMD segments*

---

**Description**

Save PMD segments in rds format and as tab-delimited file.

**Usage**

```
savePMDSegments(PMDs, GRangesFilename, TableFilename)
```

**Arguments**

PMDs                    GRanges object containing the PMD segmentation. Return value of the segmentPMDs function (see example).

GRangesFilename        Filename of the GRanges object.

TableFilename         Filename of the PMD table.

**Value**

No return value.

**Author(s)**

Lukas Burger lukas.burger@fmi.ch

**Examples**

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqlengths(Hsapiens)

# read methylation data
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
package="MethylSeekR")
meth.gr <- readMethylome(fileName=methFname, seqLengths=sLengths)

#segment PMDs
PMDsegments.gr <- segmentPMDs(m=meth.gr, chr.sel="chr22",
seqLengths=sLengths)

#save PMD segments
savePMDSegments(PMDs=PMDsegments.gr, GRangesFilename="PMDs.gr.rds",
TableFilename="PMDs.tab")
```

---

saveUMRLMRSegments	<i>Save UMR and LMR segments</i>
--------------------	----------------------------------

---

**Description**

Save UMR and LMRs segments in rds format and/or as tab-delimited file

**Usage**

```
saveUMRLMRSegments(segs, GRangesFilename = NULL, TableFilename = NULL)
```

**Arguments**

segs	GRanges object containing the UMR/LMR segmentation. Return value of the segmentUMRsLMRs function (see example).
GRangesFilename	Filename of the GRanges object.
TableFilename	Filename of the UMR/LMR table.

**Value**

No return value. Only one filename is required.

**Author(s)**

Lukas Burger lukas.burger@fmi.ch

**Examples**

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqLengths(Hsapiens)

# read methylation data
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
package="MethylSeekR")
meth.gr <- readMethylome(FileName=methFname, seqLengths=sLengths)

FDR.cutoff <- 5
m.sel <- 0.5
n.sel <- 3

#segment UMRs and LMRs, assuming no PMDs
UMRLMRsegments.gr <- segmentUMRsLMRs(m=meth.gr, meth.cutoff=m.sel,
nCpG.cutoff=n.sel, num.cores=1,
myGenomeSeq=Hsapiens, seqLengths=sLengths)

#save UMRs and LMRs
saveUMRLMRSegments(segs=UMRLMRsegments.gr, GRangesFilename="UMRsLMRs.gr.rds",
TableFilename="UMRsLMRs.tab")
```

---

`segmentPMDs`*PMD segmenter*

---

**Description**

This function trains a Hidden Markov Model (HMM) to detect partially methylated domains (PMDs) in Bis-seq data.

**Usage**

```
segmentPMDs(m, chr.sel, pdfFilename = NULL, seqLengths, num.cores = 1, nCGbin = 101)
```

**Arguments**

<code>m</code>	GRanges object containing the methylation data.
<code>chr.sel</code>	Chromosome on which HMM should be trained. Must be one of the sequence levels of <code>m</code> .
<code>pdfFilename</code>	Name of the pdf file in which the figure is saved. If no name is provided (default), the figure is printed to the screen.
<code>seqLengths</code>	A named vector indicating the chromosome lengths of the genome used.
<code>num.cores</code>	The number of cores used for the calculations (default 1).
<code>nCGbin</code>	The number of CpGs in each sliding window used to calculate alpha (default 101). The default is highly recommended.

**Value**

A GRanges object containing segments that partition the genome into PMDs and regions outside of PMDs. The object contains two metadata columns indicating the type of region (PMD/notPMD) and the number of covered (by at least 5 reads) CpGs (nCG) in the region. The function also creates a figure showing the inferred emission distributions of the HMM that is either printed to the screen (default) or saved as a pdf if a filename is provided.

**Author(s)**

Lukas Burger [lukas.burger@fmi.ch](mailto:lukas.burger@fmi.ch)

**Examples**

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqLengths(Hsapiens)

# read methylation data
```

```
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
  package="MethylSeekR")
meth.gr <- readMethylome(FileName=methFname, seqLengths=sLengths)

#segment PMDs
PMDsegments.gr <- segmentPMDs(m=meth.gr, chr.sel="chr22",
  seqLengths=sLengths)
```

segmentUMRsLMRs

*Identify UMRs and LMRs***Description**

This function identifies hypomethylated regions and classifies them into UMRs and LMRs.

**Usage**

```
segmentUMRsLMRs(m, meth.cutoff = 0.5, nCpG.cutoff = 3, PMDs = NA,
  pdfFilename, num.cores = 1, myGenomeSeq, seqLengths, nCpG.smoothing = 3,
  minCover = 5)
```

**Arguments**

<code>m</code>	GRanges object containing the methylation data.
<code>meth.cutoff</code>	Cut-off on methylation for calling hypomethylated regions.
<code>nCpG.cutoff</code>	Cut-off on the minimal number of CpGs for calling hypomethylated regions.
<code>PMDs</code>	GRanges object of PMDs. Set to either the return value of the function <code>segmentPMDs</code> (see example) or to NA if there are no PMDs (default).
<code>pdfFilename</code>	Name of the pdf file in which the figure is saved. If no name is provided (default), the figure is printed to the screen.
<code>num.cores</code>	Number of cores used for the calculations.
<code>myGenomeSeq</code>	Genome sequence as BSgenome object.
<code>seqLengths</code>	A named vector indicating the chromosome lengths of the genome used.
<code>nCpG.smoothing</code>	The number of consecutive CpGs that the methylation levels are averaged over.
<code>minCover</code>	Only CpGs with a coverage of at least <code>minCover</code> reads will be used.

**Value**

Returns a GRanges object containing all UMRs and LMRs with the following metadata values: the number of CpGs with a coverage of at least `minCover` per region (`nCG.segmentation`), the number of CpGs in the DNA sequence (`nCG`), the total number of reads that map to CpGs in the region (`T`), the total number of read that map to CpGs without conversion of the C (`M`), the mean methylation of the segment (`pmeth`), the median methylation of the segment (`median.meth`) and the type (UMR/LMR) of region (`type`). The function creates a figure showing the classification of regions into UMRs and LMRs based on the number of CpGs they contain. The figure is either printed to the screen (default) or saved as a pdf if a filename is provided.

**Author(s)**

Lukas Burger lukas.burger@fmi.ch

**Examples**

```
library(MethylSeekR)

# get chromosome lengths
library("BSgenome.Hsapiens.UCSC.hg18")
sLengths=seqlengths(Hsapiens)

# read methylation data
methFname <- system.file("extdata", "Lister2009_imr90_hg18_chr22.tab",
package="MethylSeekR")
meth.gr <- readMethylome(FileName=methFname, seqLengths=sLengths)

FDR.cutoff <- 5
m.sel <- 0.5
n.sel <- 3

#segment UMRs and LMRs, assuming no PMDs
UMRLMRsegments.gr <- segmentUMRsLMRs(m=meth.gr, meth.cutoff=m.sel,
nCpG.cutoff=n.sel, myGenomeSeq=Hsapiens, seqLengths=sLengths)
```



# Index

## \* package

MethylSeekR-package, [2](#)

calculateFDRs, [3](#)

MethylSeekR (MethylSeekR-package), [2](#)

MethylSeekR-package, [2](#)

plotAlphaDistributionOneChr, [5](#)

plotFinalSegmentation, [6](#)

plotPMDSegmentation, [7](#)

readMethylome, [9](#)

readSNPTable, [10](#)

removeSNPs, [11](#)

savePMDSegments, [12](#)

saveUMRLMRSegments, [13](#)

segmentPMDs, [14](#)

segmentUMRSLMRs, [15](#)