Package ‘DMRcate’

May 15, 2024

Title  Methylation array and sequencing spatial analysis methods
Version  3.0.0
Date  2024-04-24
Author  Tim Peters
Description  
De novo identification and extraction of differentially methylated regions (DMRs) from the human genome using Whole Genome Bisulfite Sequencing (WGBS) and Illumina Infinium Array (450K and EPIC) data. Provides functionality for filtering probes possibly confounded by SNPs and cross-hybridisation. Includes GRanges generation and plotting functions.
Depends  R (>= 4.3.0)
Imports  AnnotationHub, ExperimentHub, bsseq, GenomeInfoDb, limma, edgeR, minfi, missMethyl, GenomicRanges, plyr, Gviz, IRanges, stats, utils, S4Vectors, methods, graphics, SummarizedExperiment, biomaRt, grDevices
biocViews  DifferentialMethylation, GeneExpression, Microarray, MethylationArray, Genetics, DifferentialExpression, GenomeAnnotation, DNA Methylation, OneChannel, TwoChannel, MultipleComparison, QualityControl, TimeCourse, Sequencing, WholeGenome, Epigenetics, Coverage, Preprocessing, DataImport
License  file LICENSE
VignetteBuilder  knitr
git_url  https://git.bioconductor.org/packages/DMRcate
git_branch  RELEASE_3_19
git_last_commit  9cd77de
git_last_commit_date  2024-04-30
Repository  Bioconductor 3.19
Date/Publication  2024-05-15
Maintainer  Tim Peters <t.peters@garvan.org.au>
Description

De novo identification and extraction of differentially methylated regions (DMRs) in the human genome using Illumina array and bisulfite sequencing data. DMRcate extracts and annotates differentially methylated regions (DMRs) using a kernel-smoothed estimate. Functions are provided for filtering probes possibly confounded by SNPs and cross-hybridisation. Includes GRanges generation and plotting functions.

Author(s)

Tim J. Peters <t.peters@garvan.org.au>

References


Examples

```r
library(ExperimentHub)
library(limma)
eh <- ExperimentHub()
FlowSorted.Blood.EPIC <- eh[["EH1136"]]
tcell <- FlowSorted.Blood.EPIC[,colData(FlowSorted.Blood.EPIC)$CD4T==100 |
                          colData(FlowSorted.Blood.EPIC)$CD8T==100]
```
\begin{verbatim}
changeFDR <- function(annot, FDR) {
  remove <- apply(detP, 1, function (x) any(x > 0.01))
  tcell <- tcell[!remove,]
  tcell <- minfi::preprocessFunnorm(tcell)
  tcell <- tcell[seqnames(tcell) == "chr2",]
  tcellms <- minfi::getM(tcell)
  tcellms.noSNPs <- rmSNPandCH(tcellms, dist=2, mafcut=0.05)
  tcell$Replicate[tcell$Replicate=="""] <- tcell$Sample_Name[tcell$Replicate=="""]
  tcellms.noSNPs <- avearrays(tcellms.noSNPs, tcell$Replicate)
  tcell <- tcell[,!duplicated(tcell$Replicate)]
  colnames(tcellms.noSNPs) <- colnames(tcell)
  myannotation <- cpg.annotate("array", tcell,
      arraytype = "EPICv1", analysis.type="differential",
      design=design, design=design, coef=2)
  dmroutput <- dmcate(myannotation, lambda=1000, C=2)
  results.ranges <- extractRanges(dmroutput, genome = "hg19")
  groups <- c(CD8T="magenta", CD4T="forestgreen")
  cols <- groups[as.character(type)]
  DMR.plot(ranges=results.ranges, dmr=1,
      CpGs=minfi::getBeta(tcell), what="Beta",
      arraytype = "EPICv1", phen.col=cols, genome="hg19")
}

detP <- minfi::detectionP(tcell)
remove <- apply(detP, 1, function (x) any(x > 0.01))
tcell <- tcell[!remove,]
tcell <- minfi::preprocessFunnorm(tcell)
# Subset to chr2 only
tcell <- tcell[seqnames(tcell) == "chr2",]
tcellms <- minfi::getM(tcell)
tcellms.noSNPs <- rmSNPandCH(tcellms, dist=2, mafcut=0.05)
tcell$Replicate[tcell$Replicate=="""] <- tcell$Sample_Name[tcell$Replicate=="""]
tcellms.noSNPs <- avearrays(tcellms.noSNPs, tcell$Replicate)
tcell <- tcell[,!duplicated(tcell$Replicate)]
tcell <- tcell[rownames(tcellms.noSNPs),]
colnames(tcellms.noSNPs) <- colnames(tcell)
assays(tcell)[["M"]]<- tcellms.noSNPs
assays(tcell)[["Beta"]]<- minfi::ilogit2(tcellms.noSNPs)
type <- factor(tcell$CellType)
design <- model.matrix(~type)
myannotation <- cpg.annotate("array", tcell,
      arraytype = "EPICv1", analysis.type="differential",
      design=design, coef=2)
dmroutput <- dmcate(myannotation, lambda=1000, C=2)
results.ranges <- extractRanges(dmroutput, genome = "hg19")
groups <- c(CD8T="magenta", CD4T="forestgreen")
cols <- groups[as.character(type)]
DMR.plot(ranges=results.ranges, dmr=1,
      CpGs=minfi::getBeta(tcell), what="Beta",
      arraytype = "EPICv1", phen.col=cols, genome="hg19")
\end{verbatim}

**changeFDR**

*Change the individual CpG FDR thresholding for a CpgAnnotated object.*

**Description**

Takes a CpgAnnotated-class object and a specified FDR > 0 and < 1, and re-indexes the object in order to call DMRs at the specified rate.

**Usage**

changeFDR(annot, FDR)

**Arguments**

- annot: A CpgAnnotated-class object, created by cpg.annotate or sequencing.annotate.
- FDR: The desired individual CpG FDR, which will index the rate at which DMRs are called.
Details

The number of CpG sites called as significant by this function will set the post-smoothing threshold for DMR constituents in dmrcate.

Value

A re-indexed CpGannotated-class object.

Author(s)

Tim Peters <t.peters@garvan.org.au>

Examples

library(GenomicRanges)
stats <- rt(1000, 2)
fdrs <- p.adjust(2*pt(-abs(stats), 100), "BH")
annotated <- GRanges(rep("chr1", 1000), IRanges(1:1000, 1:1000), stat = stats, diff = 0, ind.fdr = fdrs, is.sig = fdrs < 0.05)
names(annotated) <- paste0("CpG", 1:1000)
myannotation <- new("CpGannotated", ranges=annotated)
changeFDR(myannotation, 0.1)

cpg.annotate

Annotate Illumina CpGs with their chromosome position and test statistic

Description

Annotate a matrix/GenomicRatioSet representing EPICv2, EPICv1 or 450K data with probe weights and chromosomal position. Provides replicate filtering and remapping functions for EPICv2 probes.

Usage

cpg.annotate(datatype = c("array", "sequencing"), object,
what = c("Beta", "M"), arraytype = c("EPICv2", "EPICv1", "EPIC",
"450k"), epicv2Remap = TRUE, analysis.type = c("differential",
"variability", "ANOVA", "diffVar"), design, contrasts = FALSE,
cont.matrix = NULL, fdr = 0.05, coef, varFitcoef = NULL,
topVarcoef = NULL, ...)

Arguments

datatype Character string representing the type of data being analysed.
object Either:
- A matrix of M-values, with unique Illumina probe IDs as rownames and unique sample IDs as column names or,
- A GenomicRatioSet, appropriately annotated.
what Does the data matrix contain Beta or M-values? Not needed if object is a GenomicRatioSet.

arraytype Is the data matrix sourced from EPIC or 450K data? Not needed if object is a GenomicRatioSet.

epicv2Remap Logical indicating whether to remap 11,878 cross-hybridising EPICv2 probes to their more likely CpG target (see Peters et al. 2024).

analysis.type "differential" for dmrcate() to return DMRs; "variability" to return VMRs; "ANOVA" to return "whole experiment" DMRs, incorporating all possible contrasts from the design matrix using the moderated $F$-statistics; "diffVar" to return differentially variable methylated regions, using the missMethyl package to generate $t$-statistics.

design Study design matrix. Identical context to differential analysis pipeline in limma. Must have an intercept if contrasts=FALSE. Applies only when analysis.type %in% c("differential", "ANOVA", "diffVar").

contrasts Logical denoting whether a limma-style contrast matrix is specified. Only applicable when datatype="array" and analysis.type %in% c("differential", "diffVar").

cont.matrix Limma-style contrast matrix for explicit contrasting. For each call to cpg.annotate, only one contrast will be fit. Only applicable when datatype="array" and analysis.type %in% c("differential", "diffVar").

fdr FDR cutoff (Benjamini-Hochberg) for which CpG sites are individually called as significant. Used to index default thresholding in dmrcate(). Highly recommended as the primary thresholding parameter for calling DMRs. Not used when analysis.type == "variability".

coef The column index in design corresponding to the phenotype comparison. Corresponds to the comparison of interest in design when contrasts=FALSE, otherwise must be a column name in cont.matrix. Only applicable when analysis.type == "differential".

varFitcoef The columns of the design matrix containing the comparisons to test for differential variability. If left NULL, will test all columns. Identical context to missMethyl::varFit(). Only applicable when analysis.type %in% "diffVar".

topVarcoef Column number or column name specifying which coefficient of the linear model fit is of interest. It should be the same coefficient that the differential variability testing was performed on. Default is last column of fit object. Identical context to missMethyl::topVar(). Only applicable when analysis.type %in% "diffVar".

... Extra arguments passed to the limma function lmFit() (analysis.type="differential").

Value A CpGannotated-class.

Author(s)

Tim J. Peters <ct.peters@garvan.org.au>
References


Examples

```r
library(AnnotationHub)
ah <- AnnotationHub()
EPICv2manifest <- ah[["AH116484"]]
object <- minfi::logit2(matrix(rbeta(10000, 3, 1), 1000, 10))
rownames(object) <- sample(rownames(EPICv2manifest), 1000)
type <- rep(c("Ctrl", "Treat"), each=5)
design <- model.matrix(~type)
myannotation <- cpg.annotate("array", object, what = "M", arraytype = "EPICv2",
  analysis.type="differential", design=design, coef=2)
```

---

CpGannotated-class  
An object summarising individual CpG sites fitted to a given model

Description

An S4 class that stores output from either `cpg.annotate` or `sequencing.annotate`.

Slots

- `ranges`: A GRanges object, containing CpG-level information to be passed to `dmrcate`. Mcols of this object include:
  - `stat`: Per-CpG test statistic; `t` if from `limma` or Wald if from `DSS` if using differential mode. Variance if using variability mode, `sqrt(F)` if using ANOVA mode, `t` if using `diffVar` mode.
  - `diff`: Methylation difference/coefficient. In beta space for `cpg.annotate` output and output passed from `DSS::DMLtest()`. In logit space for when a BSseq object is passed from `sequencing.annotate`. Not available for output passed from `DSS::DMLtest.multiFactor()`. Not applicable in variability, ANOVA or `diffVar` modes.
  - `ind.fdr`: False discovery rate as calculated on individual CpG sites.
  - `is.sig`: Logical determining whether a CpG site is individually significant or not. Can be adjusted using `changeFDR`. 

- `Variance`: Variance if using variability mode, `sqrt(F)` if using ANOVA mode, `t` if using `diffVar` mode.

- `diff`: Methylation difference/coefficient. In beta space for `cpg.annotate` output and output passed from `DSS::DMLtest()`. In logit space for when a BSseq object is passed from `sequencing.annotate`. Not applicable in variability, ANOVA or `diffVar` modes.

- `ind.fdr`: False discovery rate as calculated on individual CpG sites.

- `is.sig`: Logical determining whether a CpG site is individually significant or not. Can be adjusted using `changeFDR`.
Methods

CpGannotate objects have a show method that describes the data therein.

Author(s)

Tim Peters <t.peters@garvan.org.au>

---

**DMR.plot**  
*Plotting DMRs*

Description

Plots an individual DMR (in context of possibly other DMRs) as found by dmrcate. Heatmaps are shown as well as proximal coding regions, smoothed methylation values (with an option for smoothed group means) and chromosome ideogram.

Usage

```r
DMR.plot(ranges, dmr, CpGs, what = c("Beta", "M"),
          arraytype = c("EPICv2", "EPICv1", "450K"), phen.col,
          genome = c("hg19", "hg38", "mm10"), labels = names(ranges),
          group.means = FALSE, extra.ranges = NULL,
          extra.title = names(extra.ranges))
```

Arguments

- **ranges**: A GRanges object (ostensibly created by `extractRanges()`) describing DMR coordinates.
- **dmr**: Index of ranges (one integer only) indicating which DMR to be plotted.
- **CpGs**: Either:
  - A matrix of beta values for plotting, with unique Illumina probe IDs as row-names.
  - A GenomicRatioSet, annotated with the appropriate array and data types
  - A BSseq object containing per-CpG methylation and coverage counts for the samples to be plotted
- **what**: Does CpGs (if a matrix) contain Beta or M-values? Not needed if object is a GenomicRatioSet or BSseq object.
- **arraytype**: Is CpGs (if a matrix) sourced from EPIC or 450K data? Not needed if object is a GenomicRatioSet or BSseq object.
- **phen.col**: Vector of colors denoting phenotypes of all samples described in CpGs. See vignette for worked example.
- **genome**: Reference genome for annotating DMRs. Can be one of "hg19", "hg38" or "mm10"
- **labels**: Vector of DMR names to be displayed. Defaults to names(ranges).
group.means  Whether to plot smoothed methylation by individual sample (FALSE) or by groups defined in phen.col (TRUE).

extra.ranges Optional GRanges object. Will plot any range overlapping a DMR.

extra.title  Vector of names for ranges from extra.ranges. Defaults to names(extra.ranges).

Value
A plot to the current device.

Author(s)
Tim J. Peters <t.peters@garvan.org.au>, Aaron Statham <a.statham@garvan.org.au>

Examples
library(GenomicRanges)
library(AnnotationHub)
ah <- AnnotationHub()
EPICv2manifest <- ah[["AH116484"]]
dmrranges <- GRanges("chr2:86787856-86793994")
probes <- EPICv2manifest$IlmnID[EPICv2manifest$CHR=="chr2" &
  EPICv2manifest$MAPINFO > 86770000 &
  EPICv2manifest$MAPINFO < 86810000]
probes <- probes[order(EPICv2manifest$MAPINFO)]
object <- minfi::logit2(matrix(rbeta(length(probes)*10, 3, 1),
                          length(probes), 10))
rownames(object) <- probes
object[9:35, 6:10] <- minfi::logit2(matrix(rbeta(135, 1, 3), 27, 5))
cols <- c(rep("forestgreen", 5), rep("magenta", 5))
names(cols) <- rep(c("Ctrl", "Treat"), each=5)
DMR.plot(dmrranges, dmr = 1, CpGs=object, what = "M",
         arraytype="EPICv2",
         phen.col=cols, genome="hg38", group.means = TRUE)
consec = 10,
betacutoff = NULL,
min.cpgs = 2
)

Arguments

- **object**: A `CpGannotated-class`, created from `cpg.annotate` or `sequencing.annotate`.
- **lambda**: Gaussian kernel bandwidth for smoothed-function estimation. Also informs DMR bookend definition; gaps >= lambda between significant CpG sites will be in separate DMRs. Support is truncated at 5*lambda. Default is 1000 nucleotides. See details for further info.
- **C**: Scaling factor for bandwidth. Gaussian kernel is calculated where lambda/C = sigma. Empirical testing shows for both Illumina and bisulfite sequencing data that, when lambda=1000, near-optimal prediction of sequencing-derived DMRs is obtained when C is approximately 2, i.e. 1 standard deviation of Gaussian kernel = 500 base pairs. Cannot be < 0.2.
- **pcutoff**: Threshold to determine DMRs. Default implies indexing at the rate of individually significant CpGs and can be set on the `CpGannotated-class` object using `cpg.annotate`, `sequencing.annotate` or `changeFDR`. **Default highly recommended** unless you are comfortable with the risk of Type I error. If manually specified, this value will be set on the highly permissive kernel-smoothed FDR values.
- **consec**: Use `dmrcate` in consecutive mode. Treats CpG sites as equally spaced.
- **conseclambda**: Bandwidth in `CpGs` (rather than nucleotides) to use when consec=TRUE. When specified the variable lambda becomes the minimum distance separating DMRs.
- **betacutoff**: Optional filter; removes any region from the results where the absolute mean beta shift is less than the given value. Only available for Illumina array data and results produced from DSS::DMLtest().
- **min.cpgs**: Minimum number of consecutive CpGs constituting a DMR.

Details

The values of lambda and C should be chosen with care. For array data, we currently recommend that half a kilobase represent 1 standard deviation of support (lambda=1000 and C=2). If lambda is too small or C too large then the kernel estimator will not have enough support to significantly differentiate the weighted estimate from the null distribution. If lambda is too large then `dmrcate` will report very long DMRs spanning multiple gene loci, and the large amount of support will likely give Type I errors. If you are concerned about Type I errors we highly recommend using the default value of pcutoff, although this will return no DMRs if no DM CpGs are returned by limma/DSS either.

Value

A DMRResults object.
Author(s)

Tim J. Peters <t.peters@garvan.org.au>, Mike J. Buckley <Mike.Buckley@csiro.au>, Tim Triche Jr. <tim.triche@usc.edu>

References


Examples

```r
library(AnnotationHub)
library(GenomicRanges)
ah <- AnnotationHub()
EPICv2manifest <- ah[['AH116484']]chr21probes <- rownames(EPICv2manifest)[EPICv2manifest$CHR=="chr21"]coords <- EPICv2manifest[chr21probes, "MAPINFO"]stats <- rt(length(chr21probes), 2)
fdrs <- p.adjust(2*pt(-abs(stats), 100), "BH")annotated <- GRanges(rep("chr21", length(stats)), IRanges(coords, coords), stat = stats, diff = 0, ind.fdr = fdrs, is.sig = fdrs < 0.05)
names(annotated) <- chr21probes
myannotation <- new("CpGannotated", ranges=annotated)
dmrcoutput <- dmrcate(myannotation, lambda=1000, C=2)
```

DMRcate-internal *Internal DMRcate objects and functions*

Description

Internal DMRcate objects and functions
DMResults-class

Initial storage object for called DMRs - class

Description
An S4 class that stores DMR information as output from dmrcate.

Slots
This class has eight slots, summarising DMR information to be passed to extractRanges:

- coord: DMR coordinates in UCSC style.
- no.cpgs: Number of constituent CpG sites of DMR.
- min_smoothed_fdr: Minimum FDR of the smoothed estimate.
- Stouffer: Stouffer summary transform of the individual CpG FDRs.
- HMFDR: Harmonic mean of the individual CpG FDRs.
- Fisher: Fisher combined probability transform of the individual CpG FDRs.
- maxdiff: Maximum differential/coefficient within the DMR.
- meandiff: Mean differential/coefficient across the DMR.

Methods
DMResults objects have a show method describing the number of DMRs called.

Author(s)
Tim Peters <t.peters@garvan.org.au>

extractRanges

Create a GRanges object from dmrcate output.

Description
Takes a DMResults object and produces the corresponding GRanges object.

Usage
extractRanges(dmrcoutput, genome = c("hg19", "hg38", "mm10"))

Arguments
- dmrcoutput: A DMResults object.
- genome: Reference genome for annotating DMRs with promoter overlaps. Can be one of "hg19", "hg38" or "mm10". Ranges are assumed to map to the reference stated; there is no liftover.
rmPosReps

Filter out position replicates from an EPICv2 beta- or M-matrix

Description

Given a beta-value or M-value matrix with EPICv2 probe IDs as rownames, returns a truncated matrix with a 1-to-1 mapping of probe ID to CpG locus. Values returned depend on the filtering strategy selected. Replicate probes are averaged by default, but the user may optionally select individual probes per replicate group based on maximum sensitivity to methylation change or maximum precision, as per empirical cross-platform consensus testing against EPICv1 and WGBS data (Peters et al. 2024).

Usage

rmPosReps(object, filter.strategy= c("mean", "sensitivity", "precision","random"))

Arguments

object A matrix of beta- or M-values, with unique EPICv2 Illumina probe IDs as rownames.
filter.strategy

Strategy for filtering probe replicates that map to the same CpG site. "mean" takes the mean of the available probes; "sensitivity" takes the available probe most sensitive to methylation change; "precision" either selects the available probe with the lowest variation from the consensus value (most precise), or takes the mean if that confers the lowest variation instead, "random" takes a single probe at random from each replicate group.

Value

A truncated matrix with a 1-to-1 mapping of probe to CpG site. If the group mean is taken, the first probe in the group by alphabetical sorting is returned as the rowname.

Author(s)

Tim J. Peters <t.peters@garvan.org.au>

References


Examples

```r
library(ExperimentHub)
eh <- ExperimentHub()
ALLbetas <- eh["EH9451"]
ALLbetas <- ALLbetas[1:1000,]
ALLMs <- minfi::logit2(ALLbetas)
ALLMs.repmean <- rmPosReps(ALLMs, filter.strategy="mean")
```

---

**rmSNPandCH**

*Filter probes*

**Description**

Filters a matrix of M-values (or beta values) by distance to SNP/variant. Also (optionally) removes cross-hybridising probes and sex-chromosome probes. Also removes “rs” and “nv” probes from the matrix.

**Usage**

```r
rmSNPandCH(object, dist = 2, mafcut = 0.05, and = TRUE, rmcrosshyb = TRUE, rmXY = FALSE)
```
Arguments

object A matrix of M-values or beta values, with unique Illumina probe IDs as row-names.

dist Maximum distance (from CpG to SNP/variant) of probes to be filtered out. See details for when Illumina occasionally lists a CpG-to-SNP distance as being < 0.

mafcut Minimum minor allele frequency of probes to be filtered out.

If TRUE, the probe must have at least 1 SNP binding to it that satisfies both requirements in dist and mafcut for it to be filtered out. If FALSE, it will be filtered out if either requirement is satisfied. Default is TRUE.

rmcrosshyb If TRUE, filters out probes found by Peters et al. (2024) (EPICv2), Pidsley and Zotenko et al. (2016) (EPICv1) or Chen et al. (2013) (450K) to be cross-reactive with areas of the genome not at the site of interest. Default is TRUE.

rmXY If TRUE, filters out probe hybridising to sex chromosomes. Or-operator applies when combined with other 2 filters.

Details

Probes in \(-1: \text{dist}\) will be filtered out for any integer specification of dist. When a probe is listed as being “-1” nucleotides from a SNP, that SNP is immediately adjacent to the end of the probe, and is likely to confound the measurement, in addition to those listed as 0, 1 or 2 nucleotides away. See vignette for further details.

Value

A matrix, attenuated from object, with rows corresponding to probes matching user input filtered out.

Author(s)

Tim Peters <t.peters@garvan.org.au>

References


Examples

library(ExperimentHub)
eh <- ExperimentHub()
ALLbetas <- eh["EH9451"]
ALLbetas <- ALLbetas[1:1000,]
ALLMs <- minfi::logit2(ALLbetas)
ALLMs.noSNPs <- rmSNPandCH(ALLMs, rmcrosshyb = FALSE)

sequencing.annotate

Annotate a bisulfite sequencing experiment (WGBS or RRBS) with probe weights and chromosomal position.

Description

Either: - Annotate a BSseq object with chromosome position and test statistic, or - Parse output from DSS::DMLtest() or DSS::DMLtest.multiFactor() into a CpGannotated object.

Usage

sequencing.annotate(obj, methdesign, all.cov=FALSE, contrasts = FALSE, cont.matrix = NULL, fdr = 0.05, coef, ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>A BSseq object or data.frame output from DSS::DMLtest() or DSS::DMLtest.multiFactor().</td>
</tr>
<tr>
<td>methdesign</td>
<td>Methylation study design matrix describing samples and groups. Use of edgeR::modelMatrixMeth() to make this matrix is highly recommended, since it transforms a regular model.matrix (as one would construct for a microarray or RNA-Seq experiment) into a “two-channel” matrix representing methylated and unmethylated reads for each sample. Only applicable when obj is a BSseq object.</td>
</tr>
<tr>
<td>all.cov</td>
<td>If TRUE, only CpG sites where all samples have &gt; 0 coverage will be retained. If FALSE, CpG sites for which some (not all) samples have coverage=0 will be retained.</td>
</tr>
<tr>
<td>contrasts</td>
<td>Logical denoting whether a limma-style contrast matrix is specified. Only applicable when obj is a BSseq object.</td>
</tr>
<tr>
<td>cont.matrix</td>
<td>Limma-style contrast matrix for explicit contrasting. For each call to sequencing.annotate, only one contrast will be fit. Only applicable when obj is a BSseq object.</td>
</tr>
<tr>
<td>fdr</td>
<td>FDR cutoff (Benjamini-Hochberg) for which CpG sites are individually called as significant. Used to index default thresholding in dmrcate(). Highly recommended as the primary thresholding parameter for calling DMRs. Only applicable when obj is a BSseq object.</td>
</tr>
<tr>
<td>coef</td>
<td>The column index in design corresponding to the phenotype comparison. Corresponds to the comparison of interest in design when contrasts=FALSE, otherwise must be a column name in cont.matrix. Only applicable when obj is a BSseq object.</td>
</tr>
<tr>
<td>...</td>
<td>Extra arguments passed to the limma function lmFit(). Only applicable when obj is a BSseq object.</td>
</tr>
</tbody>
</table>
Value

A \texttt{CpGannotated-class}.

Author(s)

Tim J. Peters <t.peters@garvan.org.au>

References


Examples

```r
library(ExperimentHub)
library(SummarizedExperiment)
library(bsseq)
library(GenomeInfoDb)
eh = ExperimentHub()
bis_1072 <- eh[["EH1072"]]
pData(bis_1072) <- data.frame(replicate=gsub(".*-", "", colnames(bis_1072)),
                              tissue=substr(colnames(bis_1072), 1, nchar(colnames(bis_1072))-3),
                              row.names=colnames(bis_1072))
colData(bis_1072)$tissue <- gsub("-", "_", colData(bis_1072)$tissue)
bis_1072 <- renameSeqlevels(bis_1072, mapSeqlevels(seqlevels(bis_1072), "UCSC"))
bis_1072 <- bis_1072[seqnames(bis_1072)="chr19",]
bis_1072 <- bis_1072[240201:240300,]
tissue <- factor(pData(bis_1072)$tissue)
tissue <- relevel(tissue, "Liver_Treg")
design <- model.matrix(~tissue)
colnames(design) <- gsub("tissue", "", colnames(design))
colnames(design)[1] <- "Intercept"
rownames(design) <- colnames(bis_1072)
methdesign <- edgeR::modelMatrixMeth(design)
cont.mat <- limma::makeContrasts(treg_vs_tcon=Lymph_N_Treg-Lymph_N_Tcon, 
                               fat_vs_ln=Fat_Treg-Lymph_N_Treg, 
                               skin_vs_ln=Skin_Treg-Lymph_N_Treg, 
                               fat_vs_skin=Fat_Treg-Skin_Treg, 
                               levels=methdesign)
seq_annot <- sequencing.annotate(bis_1072, methdesign, all.cov = TRUE, 
                                 contrasts = TRUE, cont.matrix = cont.mat, 
                                 coef = "treg_vs_tcon", fdr=0.05)
```
**Index**

* classes
  - CpGannotated-class, 6
  - DMResults-class, 11

* internal
  - DMRcate-internal, 10

changeFDR, 3, 6, 9

cpg.annotate, 3, 4, 6, 9

CpGannotated-class, 6

DMR.plot, 7

DMRcate (DMRcate-package), 2
dmrcate, 4, 6, 8, 11

DMRcate-internal, 10

DMRcate-package, 2

DMResults-class, 11

extractCoords (DMRcate-internal), 10

extractRanges, 11, 11

fitParallel (DMRcate-internal), 10

KernelSums (DMRcate-internal), 10

KernelTest (DMRcate-internal), 10

plot (DMR.plot), 7

rmPosReps, 12

rmSNPandCH, 13

Segment (DMRcate-internal), 10

sequencing.annotate, 3, 6, 9, 15

SparseDeltas (DMRcate-internal), 10