Package ‘DMRcate’

June 12, 2022

Title Methylation array and sequencing spatial analysis methods

Version 2.10.0

Date 2022-07-02

Author Tim Peters

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

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 (>= 4.0.0)

Imports ExperimentHub, bsseq, GenomeInfoDb, limma, edgeR, DSS, minfi, missMethyl, GenomicRanges, plyr, Gviz, IRanges, stats, utils, S4Vectors, methods, graphics, SummarizedExperiment

biocViews DifferentialMethylation, GeneExpression, Microarray, MethylationArray, Genetics, DifferentialExpression, GenomeAnnotation, DNA Methylation, OneChannel, TwoChannel, MultipleComparison, QualityControl, TimeCourse, Sequencing, Whole Genome, Epigenetics, Coverage, Preprocessing, DataImport


License file LICENSE

VignetteBuilder knitr

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

git_branch RELEASE_3_15

git_last_commit 81e8370

git_last_commit_date 2022-04-26

Date/Publication 2022-06-12
**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]
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
```
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.
cpg.annotate

Annotate Illumina Cpgs with their chromosome position and test statistic

description

Annotate a matrix/GenomicRatioSet representing 450K or EPIC data with probe weights and chromosomal position.

Usage

cpg.annotate(datatype = c("array", "sequencing"), object, what=c("Beta", "M"), arraytype=c("EPIC", "450K"), analysis.type = c("differential",

Examples

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[138151:138250,]
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)
methoddesign <- 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=methoddesign)
seq_annot <- sequencing.annotate(bis_1072, methoddesign, all.cov = TRUE,
  contrasts = TRUE, cont.matrix = cont.mat,
  coef = "fat_vs skin", fdr=0.05)
seq_annot <- changeFDR(seq_annot, 0.25)
"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 Ge-
  nomicRatioSet.

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

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 applic-
  cable 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 recom-
  mended 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. Cor-
  responds to the comparison of interest in design when contrasts=FALSE, oth-
  erwise 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 dif-
  ferential 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. Identi-
  cal context to missMethyl::topVar(). Only applicable when analysis.type
  %in% "diffVar".

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

A \textit{CpGannotated-class}.

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]
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$Replicate[!duplicated(tcell$Replicate)]
tcell$Sample_Name[!duplicated(tcell$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 = "EPIC",
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`.

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 group means and chromosome ideogram.

Usage

```
DMR.plot(ranges, dmr, CpGs, what=c("Beta", "M"),
arraytype=c("EPIC", "450K"), phen.col,
genome = c("hg19", "hg38", "mm10"), ...)
```
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"
... Extra arguments passed to Gviz:::plotTracks().

Value

A plot to the current device.

Author(s)

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

Examples

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]
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$Replicate[tcCell$Replicate==""| tcell$Sample_Name[tcCell$Replicate==""] <- tcell$Sample_Name[tcCell$Replicate=="""]
tcellms.noSNPs <- avearrays(tcellms.noSNPs, tcell$Replicate)
tcell <- tcell[,!duplicated(tcell$Replicate)]
The main function of this package. Computes a kernel estimate against a null comparison to identify significantly differentially (or variable) methylated regions.

Usage

```r
dmrcate(object,
    lambda = 1000,
    C=NULL,
    pcutoff = "fdr",
    consec = FALSE,
    conseclambda = 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.

consec.lambda
Bandwidth in `CpGs` (rather than nucleotides) to use when `consec=TRUE`. When specified the variable `lambda` simply 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 DMResults 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(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]
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
rnames(tcell) == "chr2",]
tcellms <- minfi::getM(tcell)
tcellms.noSNPs <- rmSNPandCH(tcellms, dist = 2, mafcut = 0.05)
tcell$Replicate[!is.na(tcell$Replicate)] <- tcell$Sample_Name[!is.na(tcell$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 = "EPIC",
                         analysis.type="differential", design=design, coef=2)
dmrcoutput <- dmrcate(myannotation, lambda=1000, C=2)
```

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.

Value

A GRanges object.

Author(s)

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

Examples

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]
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 = "EPIC",
                        analysis.type="differential", design=design, coef=2)
dmrcoutput <- dmrcate(myannotation, lambda=1000, C=2)
results.ranges <- extractRanges(dmrcoutput, genome = "hg19")

---

**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.

**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.
- **and**: 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 Pidsley and Zotenko et al. (2016) for EPIC or Chen et al. (2013) for 450K to be cross-reactive with areas of the genome not at the site of interest. Many of these sites are on the X-chromosome, leading to potential confounding if the sample group is a mix of males and females. There are 63,707 probes in total in this list. 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:dist will be filtered out for any integer specification of dist. When a probe is listed as being “-1” nucleotides from a SNP (7 in total of the 153,113), 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 J. Peters <t.peters@garvan.org.au>

References


Examples

```r
library(ExperimentHub)
library(limma)
eh <- ExperimentHub()
FlowSorted.Blood.EPIC <- eh["EH1136"]
detP <- minfi::detectionP(tcell)
remove <- apply(detP, 1, function(x) any(x > 0.01))
tcell <- tcell[!remove,]
tcell <- minfi::preprocessFunnorm(tcell)
tcellms <- minfi::getM(tcell)
tcellms.noSNPs <- rmSNPandCH(tcellms, dist=2, mafcut=0.05)
```

sequencing.annotate

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

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

obj A BSseq object or data.frame output from DSS::DMLtest() or DSS::DMLtest.multiFactor().
methdesign 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.
all.cov If TRUE, only CpG sites where all samples have > 0 coverage will be retained. If FALSE, CpG sites for which some (not all) samples have coverage=0 will be retained.
contrasts Logical denoting whether a limma-style contrast matrix is specified. Only applicable when obj is a BSseq object.
cont.matrix 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.
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. Only applicable when obj is a BSseq object.
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 obj is a BSseq object.

Value

A CpGannotated-class.

Author(s)

Tim J. Peters <ct.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, 7
  DMRResults-class, 11
changeFDR, 3, 7, 10
cpg.annotate, 3, 4, 7, 9, 10
CpGannotated-class, 7
DMR.plot, 7
DMRcate (DMRcate-package), 2
dmrcate, 3, 7, 9, 11, 12
DMRcate-package, 2
DMRResults-class, 11
extractRanges, 11, 12
plot (DMR.plot), 7
rmSNPandCH, 13
sequencing.annotate, 3, 7, 9, 10, 14