Package: methylTFR (via r-universe)

November 20, 2026

Title Quantification of DNA Methylation Signatures in TFBS

Version 0.99.12

Date 2026-11-20

Description methylTFR is an extensive R package designed specifically for analyzing DNA methylation signatures within transcription factor binding sites (TFBS) across diverse genomic annotations. It enhances methylation analysis by calculating a bias-corrected 'deviation' in methylation levels, leveraging GC bias distribution and frequency data. This package offers a variety of functions that facilitate the motif activity exhibiting diverse methylation patterns across distinct samples or cells.

License MIT + file LICENSE

URL https://github.com/EpigenomeInformatics/methylTFR, https://epigenomeinformatics.github.io/methylTFR/

BugReports https://support.bioconductor.org/t/methylTFR

Depends data.table (>= 1.14.0), R (>= 4.3.0), SummarizedExperiment

Imports BiocGenerics, BiocParallel, DelayedArray, GenomicRanges, ggplot2, HDF5Array, IRanges, logger, matrixStats, methods, R.utils, S4Vectors, stats, stringr, utils

Suggests BiocStyle, circlize, ComplexHeatmap, grid, knitr, RnBeads, RnBeads.hg38, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

biocViews Software, DNAMethylation, SingleCell, StatisticalMethod, GeneRegulation, Epigenetics,Transcription, Sequencing, WholeGenome, DifferentialMethylation, MotifAnnotation, MotifDiscovery, Coverage

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs libicu-dev libssl-dev zlib1g-dev

Repository https://bioc.r-universe.dev

Date/Publication 2026-11-20 12:53:57 UTC

RemoteUrl https://github.com/bioc/methylTFR

RemoteRef HEAD

RemoteSha 67c014612aba51afa77a76a4286452f32f3bcecd

Contents

addGCBinToMethylome 2
cbind,methylTFRdeviations-method 3
computeDeviation 4
computeZScoreVariability 6
deviations 8
deviations,methylTFRdeviations-method 8
deviationZScores 9
deviationZScores,methylTFRdeviations-method 10
differentialDeviationTest 10
methylTFRdeviations-class 12
plotExpectedFootprint 12
plotMotifFootprint 14
rbind,methylTFRdeviations-method 15
readMethylome 16
runMethylTFR 17
runMethylTFRFromRnBeads 19

Index **23**

`addGCBinToMethylome` *addGCBinToMethylome*

Description

Assign each methylation site to the GC bin of the genomic window it falls in, and compute the average methylation per GC bin.

Usage

```
addGCBinToMethylome(msites, gcdist, ignoreStrand = TRUE)
```

Arguments

msites	A GRanges object with one range per CpG (width 1) and the metadata columns score (numeric methylation fraction between 0 and 1) and coverage (read count). Usually produced by readMethylome.
gcdist	A GRanges object of 30-bp genomic windows with an integer metadata column GC_bin (1 to 5). Get it with getGenomeGC() from the annotation package.
ignoreStrand	Logical. If TRUE (default), strand is ignored when overlapping methylation sites with annotations.

Details

The result describes how methylation depends on local GC content in this sample. computeDeviation combines it with the per-motif GC profile in gcfreqs to obtain the methylation expected at the binding sites from sequence composition alone.

Value

A matrix with one row per GC bin and the columns gcbin and avg_mscore. Pass it as binMsites to computeDeviation.

Author(s)

Irem Gunduz

See Also

computeDeviation, readMethylome

Examples

```
load(system.file("extdata", "gcdist_subset.rda", package = "methylTFR"))
load(system.file("extdata", "example_data.rda", package = "methylTFR"))

bin_meth <- addGCBinToMethylome(msites, gcdist)
bin_meth
```

---

cbind,methylTFRdeviations-method
cbind

Description

Combine methylTFRdeviations objects by column.

Usage

```
## S4 method for signature 'methylTFRdeviations'
cbind(..., deparse.level = 1)
```

Arguments

`...`	methylTFRdeviations objects.
`deparse.level`	passed to the underlying method.

Details

Defined explicitly rather than left to inheritance. Although `methylTFRdeviations` extends `SummarizedExperiment`, whether that class's `cbind` method is found for a subclass defined in another package depends on the search path and on the Bioconductor version: dispatch can fall through to the `S4Vectors` method, which then fails with "unable to find an inherited method for function 'bindCOLS'".

The objects are therefore coerced to `SummarizedExperiment` explicitly, combined there, and re-wrapped. This is defined with `setMethod` on the existing generic rather than `setGeneric`, so it does not shadow the base function and any number of objects can be combined.

Value

A methylTFRdeviations object.

Examples

```
load(system.file("extdata", "tc_mem.rda", package = "methylTFR"))
load(system.file("extdata", "tc_naive.rda", package = "methylTFR"))
devs <- cbind(tc_mem, tc_naive)
```

`computeDeviation` *computeDeviation*

Description

Compute the bias-corrected deviation score of one motif in one sample.

Usage

```
computeDeviation(
  motif,
  msites,
  tf_bindsites,
  gcfreqs,
  enhancer = NULL,
  ignoreStrand = TRUE,
  binMsites,
  extendBy = 130,
  windowBreaks = c(-250, -200, -25, 25, 200, 250)
)
```

Arguments

motif	Name of the motif, one of `names(tf_bindsites)`.
msites	A `GRanges` object with one range per CpG (width 1) and the metadata columns `score` (numeric methylation fraction between 0 and 1) and `coverage` (read count). Usually produced by `readMethylome`.
tf_bindsites	A `GRangesList` named by motif. Each element is a `GRanges` of the occurrences of that motif. The names must include every name in `names(gcfreqs)`. Get it with `getTFbindsites(motifSet = "jaspar2020")` from the **methylTFRAnnotationHg38** (or **methylTFRAnnotationMm10**) package, or build a custom set with **methylTFRAnnotationBuilder**.
gcfreqs	A named `list` with one element per motif, named as in `tf_bindsites`. Each element is a numeric matrix with the five GC bins in rows and the positions across the binding-site window in columns. Get it with `getGCfreq(motifSet = ...)` from the annotation package.
enhancer	An optional `GRanges` object of regions, for example distal regulatory elements. Binding sites outside these regions are dropped. Any region set imported as `GRanges` works, for example the ENCODE candidate cis-regulatory elements (SCREEN) read with `rtracklayer::import()`. Default `NULL` uses all binding sites.
ignoreStrand	Logical. If `TRUE` (default), strand is ignored when overlapping methylation sites with annotations.
binMsites	The matrix returned by `addGCBinToMethylome`, with the columns `gcbin` and `avg_mscore`.
extendBy	Number of base pairs added to the width of each binding site before overlapping it with the methylation sites, split evenly between both sides. Default 130.
windowBreaks	Six strictly increasing distances from the site centre, in base pairs, that define the five distance bins. Bins 1 and 5 are the outer flanks and bin 3 is the centre. Default `c(-250, -200, -25, 25, 200, 250)`.

Details

Each binding site is widened by `extendBy` base pairs and every CpG inside it is placed by its distance to the site centre. The distances are cut into five bins at `windowBreaks`: an outer left flank, an inner left region, the centre, an inner right region and an outer right flank. The observed score is the mean methylation in the centre bin divided by the mean of the two outer flank bins. The same ratio is computed for the methylation expected from GC content (see `addGCBinToMethylome`), and the deviation is the observed minus the expected ratio. If the centre or either outer flank bin holds no CpG, the score is `NA`.

Every CpG counts once, whatever its read coverage: the bin means are unweighted means of the per-CpG methylation fractions. Coverage only enters through the `cov_threshold` filter applied when the data are read.

Value

A `data.table` with one row and the columns `dev` (bias-corrected deviation) and `exp_dev` (expected deviation).

See Also

runMethylTFR to score all motifs and samples at once, plotMotifFootprint to see the profile behind the score.

Examples

```
load(system.file("extdata", "BATF_tf_bindsites.rda", package = "methylTFR"))
load(system.file("extdata", "example_data.rda", package = "methylTFR"))
load(system.file("extdata", "BATF_gcfreqs.rda", package = "methylTFR"))
load(system.file("extdata", "gcdist_subset.rda", package = "methylTFR"))

bin_meth <- addGCBinToMethylome(msites, gcdist, TRUE)

computeDeviation(
    motif = "BATF",
    msites = msites,
    tf_bindsites = tf_bindsites,
    gcfreqs = gcfreqs,
    enhancer = NULL,
    ignoreStrand = TRUE,
    binMsites = bin_meth
)
```

---

computeZScoreVariability

computeZScoreVariability

Description

Identify transcription factor motifs whose methylTFR activity varies across samples more than expected by chance.

Usage

```
computeZScoreVariability(
  object,
  method = c("robust", "gaussian"),
  bootstrap = FALSE,
  niterations = 1000L,
  conf_level = 0.95,
  padjMethod = "BH"
)
```

Arguments

object	A methylTFRdeviations object, or a numeric matrix or data.frame of bias-corrected deviation scores with motifs in rows and samples in columns.

method	Either "robust" (median and median absolute deviation, the default) or "gaussian" (mean and standard deviation), used to estimate the within-sample null.
bootstrap	logical, if TRUE bootstrap confidence bounds for the variability estimates are computed by resampling samples with replacement.
niterations	Number of bootstrap iterations, used only when bootstrap is TRUE.
conf_level	Confidence level for the bootstrap bounds.
padjMethod	Method for p-value adjustment, passed to p.adjust. Default is "BH".

Details

Variability is defined as the standard deviation, across samples, of calibrated deviation scores. Calibration is what makes the number interpretable: deviation scores are first standardised against a within-sample null estimated across all motifs (see calibrateDeviations), so that a calibrated score of 1 corresponds to one unit of sample-level noise. A motif with variability greater than 1 therefore varies across samples by more than the background spread, and variability near 1 is what a motif carrying no signal is expected to show.

Under the null hypothesis that a motif's calibrated scores are independent draws from a standard normal distribution, $(n-1)s^2$ follows a chi-squared distribution with $n-1$ degrees of freedom, where s is the observed variability and n the number of samples with a non-missing score for that motif. P-values are obtained from the upper tail of that distribution and adjusted for multiple testing across motifs.

Note that this function must not be applied to the z assay returned by deviationZScores. Those are row-wise Z-scores computed for visualisation, and their standard deviation across samples is 1 for every motif by construction, which would make the test degenerate. The default input is therefore the deviations assay.

The approach is analogous to chromVAR::computeVariability, with one difference: chromVAR calibrates each motif against a set of GC- and width-matched background peak sets, whereas here the null is estimated across motifs within each sample. The latter needs no additional deviation computations and can be applied to existing results, at the cost of assuming that most motifs are inactive in any given sample.

Value

A data.frame with one row per motif and the columns motifs, variability, p_value and p_value_adjusted, plus bootstrap_lower_bound and bootstrap_upper_bound when bootstrap is TRUE. Rows are returned in the order of the input.

Author(s)

Irem Gunduz

See Also

differentialDeviationTest for testing activity differences between predefined groups.

Examples

```
# Load example data
load(system.file("extdata", "tc_mem.rda", package = "methylTFR"))
load(system.file("extdata", "tc_naive.rda", package = "methylTFR"))
devs <- cbind(tc_mem, tc_naive)

# Rank motifs by how much their activity varies across samples
var_res <- computeZScoreVariability(devs)
head(var_res[order(-var_res$variability), ])
```

---

deviations *deviations*

Description

Function to get deviations from a methylTFRdeviations object.

Usage

```
deviations(x)
```

Arguments

x	A methylTFRdeviations object.

Value

A matrix of deviations.

Examples

```
# Load example data
load(system.file("extdata", "tc_mem.rda", package = "methylTFR"))
# Get deviations
deviations(tc_mem)
```

---

deviations,methylTFRdeviations-method *deviations*

Description

Extract bias corrected deviations from methylTFRdeviations object.

Usage

```
## S4 method for signature 'methylTFRdeviations'
deviations(x)
```

Arguments

x	methylTFRdeviations object.

Value

A matrix of bias corrected deviations.

Examples

```
# Load the data
load(system.file("extdata", "tc_mem.rda", package = "methylTFR"))
# Get deviations
deviations(tc_mem)
```

---

deviationZScores *deviationZScores*

Description

Function to get deviation Z-scores from a methylTFRdeviations object.

Usage

```
deviationZScores(x)
```

Arguments

x	A methylTFRdeviations object.

Value

A matrix of deviation Z-scores.

Examples

```
# Load example data
load(system.file("extdata", "tc_mem.rda", package = "methylTFR"))
# Get deviation Z-scores
deviationZScores(tc_mem)
```

deviationZScores,methylTFRdeviations-method *deviationZScores*

Description

Extract deviation Z-scores from methylTFRdeviations object.

Usage

```
## S4 method for signature 'methylTFRdeviations'
deviationZScores(x)
```

Arguments

x	methylTFRdeviations object.

Value

A matrix of deviation Z-scores.

Examples

```
# Load the data
load(system.file("extdata", "tc_mem.rda", package = "methylTFR"))
# Get deviation Z-scores
deviationZScores(tc_mem)
```

differentialDeviationTest *differentialDeviationTest*

Description

Test which motifs have different deviation scores between groups of samples. Adapted from chromVAR::differentialDevia

Usage

```
differentialDeviationTest(
  deviations,
  groups = NULL,
  motifs = rownames(deviations),
  alternative = c("two.sided", "less", "greater"),
  parametric = TRUE,
  padjMethod = "BH"
)
```

Arguments

deviations	A methylTFRdeviations object, or a matrix or data.frame of deviation scores with motifs in rows and samples in columns.
groups	A vector with one group label per sample. If NULL, the column names of deviations are used. For two groups, the factor levels set the direction of mean_difference.
motifs	Motif names, by default rownames(deviations).
alternative	The alternative hypothesis of the two-group tests, one of "two.sided", "less" or "greater".
parametric	Logical. If TRUE (default), parametric tests are used, otherwise non-parametric tests.
padjMethod	Method for p-value adjustment, passed to p.adjust. Default "BH".

Details

With two groups, a Welch t-test (parametric = TRUE) or a Wilcoxon rank sum test is used; with more groups, a one-way ANOVA or a Kruskal-Wallis test.

Value

A data.frame with one row per motif and the columns motifs, p_value, p_value_adjusted and mean_difference. For two groups, mean_difference is the mean of the first group minus the mean of the second; for more groups, it is the range of the group means.

See Also

computeZScoreVariability for ranking motifs without group labels.

Examples

```
load(system.file("extdata", "tc_mem.rda", package = "methylTFR"))
load(system.file("extdata", "tc_naive.rda", package = "methylTFR"))
devs <- cbind(tc_mem, tc_naive)

# Construct group labels from sample names
groups <- sub("\\.bedGraph.*$", "", vapply(
    strsplit(colnames(devs), split = "_"), `[`, character(1), 1L
))

tc_result <- differentialDeviationTest(
    deviations = devs,
    groups = groups,
    alternative = "two.sided",
    parametric = TRUE,
    padjMethod = "BH"
)
head(tc_result)
```

`methylTFRdeviations-class` *methylTFRdeviations*

Description

Class for storing results from `runMethylTFR` and `runMethylTFRFromRnBeads`.

Details

This class inherits from `SummarizedExperiment`, and most methods for that class should work for objects of this class as well. Additionally, two accessor functions are defined for extracting bias corrected deviations (`deviations`) and deviation Z-scores (`deviationZScores`).

`plotExpectedFootprint` *plotExpectedFootprint*

Description

Plot the observed methylation profile around the binding sites of one motif together with the profile expected from GC content.

Usage

```
plotExpectedFootprint(
  motif,
  tf_bindsites,
  msites,
  sample_name = NULL,
  gc_dist,
  gcfreqs,
  enhancer = NULL,
  returnPlotData = FALSE
)
```

Arguments

`motif`	Name of the motif, one of `names(tf_bindsites)`.
`tf_bindsites`	A `GRangesList` named by motif. Each element is a `GRanges` of the occurrences of that motif. The names must include every name in `names(gcfreqs)`. Get it with `getTFbindsites(motifSet = "jaspar2020")` from the **methylTFRAnnotationHg38** (or **methylTFRAnnotationMm10**) package, or build a custom set with **methylTFRAnnotationBuilder**.

msites	A GRanges object with one range per CpG (width 1) and the metadata columns score (numeric methylation fraction between 0 and 1) and coverage (read count). Usually produced by readMethylome.
sample_name	Optional sample label used in the plot title. Default "sample".
gc_dist	A GRanges object of 30-bp genomic windows with an integer metadata column GC_bin (1 to 5). Get it with getGenomeGC() from the annotation package.
gcfreqs	A named list with one element per motif, named as in tf_bindsites. Each element is a numeric matrix with the five GC bins in rows and the positions across the binding-site window in columns. Get it with getGCfreq(motifSet = ...) from the annotation package.
enhancer	An optional GRanges object of regions, for example distal regulatory elements. Binding sites outside these regions are dropped. Any region set imported as GRanges works, for example the ENCODE candidate cis-regulatory elements (SCREEN) read with rtracklayer::import(). Default NULL uses all binding sites.
returnPlotData	Logical. If TRUE, return the plot data as well as the plot.

Value

A ggplot object, or, if returnPlotData is TRUE, a list with the elements plot and plotDF.

See Also

plotMotifFootprint for the bias-corrected profile, computeDeviation for the score it summarises.

Examples

```
load(system.file("extdata", "BATF_tf_bindsites.rda", package = "methylTFR"))
load(system.file("extdata", "example_data.rda", package = "methylTFR"))
load(system.file("extdata", "BATF_gcfreqs.rda", package = "methylTFR"))
load(system.file("extdata", "gcdist_subset.rda", package = "methylTFR"))

p <- plotExpectedFootprint(
    motif = "BATF",
    tf_bindsites = tf_bindsites,
    msites = msites,
    sample_name = "Sample1",
    gc_dist = gcdist,
    gcfreqs = gcfreqs,
    enhancer = NULL,
    returnPlotData = FALSE
)
```

plotMotifFootprint *plotMotifFootprint*

Description

Plot the bias-corrected methylation profile around the binding sites of one motif in one sample.

Usage

```
plotMotifFootprint(
  motif,
  tf_bindsites,
  msites,
  sample_name = NULL,
  gc_dist,
  gcfreqs,
  enhancer = NULL,
  method = "division",
  flankNorm = 50
)
```

Arguments

motif	Name of the motif, one of `names(tf_bindsites)`.
tf_bindsites	A `GRangesList` named by motif. Each element is a `GRanges` of the occurrences of that motif. The names must include every name in `names(gcfreqs)`. Get it with `getTFbindsites(motifSet = "jaspar2020")` from the **methylTFRAnnotationHg38** (or **methylTFRAnnotationMm10**) package, or build a custom set with **methylTFRAnnotationBuilder**.
msites	A `GRanges` object with one range per CpG (width 1) and the metadata columns `score` (numeric methylation fraction between 0 and 1) and `coverage` (read count). Usually produced by `readMethylome`.
sample_name	Optional sample label used in the plot title. Default `"sample"`.
gc_dist	A `GRanges` object of 30-bp genomic windows with an integer metadata column `GC_bin` (1 to 5). Get it with `getGenomeGC()` from the annotation package.
gcfreqs	A named `list` with one element per motif, named as in `tf_bindsites`. Each element is a numeric matrix with the five GC bins in rows and the positions across the binding-site window in columns. Get it with `getGCfreq(motifSet = ...)` from the annotation package.
enhancer	An optional `GRanges` object of regions, for example distal regulatory elements. Binding sites outside these regions are dropped. Any region set imported as `GRanges` works, for example the ENCODE candidate cis-regulatory elements (SCREEN) read with `rtracklayer::import()`. Default `NULL` uses all binding sites.

`method`	How observed and expected profiles are combined, either `"substraction"` (observed minus expected) or `"division"` (observed over expected, the default).
`flankNorm`	Width in base pairs of the outer flanking window used to normalise the curve. Default 50; `0` turns normalisation off.

Value

A `ggplot` object.

See Also

`plotExpectedFootprint` for the two profiles before correction.

Examples

```
load(system.file("extdata", "BATF_tf_bindsites.rda", package = "methylTFR"))
load(system.file("extdata", "example_data.rda", package = "methylTFR"))
load(system.file("extdata", "BATF_gcfreqs.rda", package = "methylTFR"))
load(system.file("extdata", "gcdist_subset.rda", package = "methylTFR"))

p <- plotMotifFootprint(
    motif = "BATF",
    tf_bindsites = tf_bindsites,
    msites = msites,
    sample_name = "Sample1",
    gc_dist = gcdist,
    gcfreqs = gcfreqs,
    enhancer = NULL,
    method = "division"
)
```

`rbind,methylTFRdeviations-method` *rbind*

Description

Combine methylTFRdeviations objects by row.

Usage

```
## S4 method for signature 'methylTFRdeviations'
rbind(..., deparse.level = 1)
```

Arguments

`...`	methylTFRdeviations objects.
`deparse.level`	passed to the underlying method.

Details

See cbind for why this is defined explicitly rather than inherited.

Value

A methylTFRdeviations object.

Examples

```
load(system.file("extdata", "tc_mem.rda", package = "methylTFR"))
devs <- rbind(tc_mem[seq_len(2), ], tc_mem[3:4, ])
```

---

readMethylome *readMethylome*

Description

Import single-sample methylation calls into a GRanges object.

Usage

```
readMethylome(filename, type, cov_threshold = 1)
```

Arguments

filename	Path to the methylation call file. Gzipped files are read directly.
type	Format of the file, one of "epp", "bissnp", "bismarkCytosine", "bismarkCov", "allc" or "encode" (ENCODE bedMethyl). Case-insensitive.
cov_threshold	Numeric. Sites with fewer reads than this are dropped. Default 1.

Value

A GRanges object with one range per CpG and the metadata columns score (methylation fraction between 0 and 1) and coverage (read count). This is the msites input of computeDeviation, addGCBinToMethylome and the footprint plots.

Author(s)

Irem Gunduz

See Also

runMethylTFR, which calls this function for every sample.

Examples

```
# Read bissnp file
bissnp_path <- system.file(
    "extdata",
    "bissnp.tsv.gz",
    package = "methylTFR"
)
bissnp <- readMethylome(bissnp_path, "bissnp")
```

---

runMethylTFR *runMethylTFR*

Description

Compute bias-corrected deviation scores for every motif and every sample listed in a sample annotation file.

Usage

```
runMethylTFR(
  sample_ann,
  sample_dir,
  tf_bindsites = NULL,
  gcfreqs = NULL,
  gc_dist = NULL,
  sampleColName = "bedFile",
  chunkSize = 20,
  full_path = FALSE,
  annfile = NULL,
  threads = 1,
  enhancer = NULL,
  filetype = NULL,
  ignoreStrand = TRUE,
  cov_threshold = 1,
  tempDir = file.path(tempdir(), "methylTFR")
)
```

Arguments

sample_ann	Name of a tab-separated (.tsv) or comma-separated (.csv) sample annotation file in sample_dir. It must have a header and contain the column named by sampleColName, which holds the methylation file names. All columns are kept as colData of the result. Not used when annfile is given.
sample_dir	Directory holding the methylation call files (and, usually, the sample annotation file).

tf_bindsites	A GRangesList named by motif. Each element is a GRanges of the occurrences of that motif. The names must include every name in names(gcfreqs). Get it with getTFbindsites(motifSet = "jaspar2020") from the **methylTFRAnnotationHg38** (or **methylTFRAnnotationMm10**) package, or build a custom set with **methylTFRAnnotationBuilder**.
gcfreqs	A named list with one element per motif, named as in tf_bindsites. Each element is a numeric matrix with the five GC bins in rows and the positions across the binding-site window in columns. Get it with getGCfreq(motifSet = ...) from the annotation package.
gc_dist	A GRanges object of 30-bp genomic windows with an integer metadata column GC_bin (1 to 5). Get it with getGenomeGC() from the annotation package.
sampleColName	Name of the column in sample_ann that holds the methylation file names. Default "bedFile".
chunkSize	Number of motifs processed together in one block. Default 20.
full_path	Logical. If TRUE, the file names in sampleColName are full paths and sample_dir is not prepended. Default FALSE.
annfile	Optional path to the sample annotation file. If given, it is used instead of sample_ann in sample_dir.
threads	Number of workers used to process the motifs of a chunk in parallel. Default 1 (serial).
enhancer	An optional GRanges object of regions, for example distal regulatory elements. Binding sites outside these regions are dropped. Any region set imported as GRanges works, for example the ENCODE candidate cis-regulatory elements (SCREEN) read with rtracklayer::import(). Default NULL uses all binding sites.
filetype	Format of the methylation call files, one of "bissnp", "epp", "allc", "bismarkCytosine", "bismarkCov" or "encode" (case-insensitive). See readMethylome.
ignoreStrand	Logical. If TRUE (default), strand is ignored when overlapping methylation sites with annotations.
cov_threshold	Numeric. Sites with fewer reads than this are dropped. Default 1.
tempDir	Directory for the temporary HDF5 files written during a run. Default is a folder under tempdir(). The files are deleted when the run finishes.

Details

Each sample is read with readMethylome, binned by GC content with addGCBinToMethylome and scored per motif with computeDeviation.

Value

A methylTFRdeviations object with the assays deviations, z and expected, motifs in rows and samples in columns.

See Also

runMethylTFRFromRnBeads for running methylTFR directly on a preprocessed RnBeads object, deviations and deviationZScores to extract the scores.

Examples

```
# A minimal end-to-end run on the BATF example data bundled with the
# package. The annotation objects cover a single motif, so the result
# has one row.
load(system.file("extdata", "example_data.rda", package = "methylTFR"))
load(system.file("extdata", "BATF_tf_bindsites.rda", package = "methylTFR"))
load(system.file("extdata", "BATF_gcfreqs.rda", package = "methylTFR"))
load(system.file("extdata", "gcdist_subset.rda", package = "methylTFR"))

# runMethylTFR() reads per-sample calls from disk, so the bundled sites
# are written out as a bismarkCov file first.
sample_dir <- tempfile("methylTFR_example")
dir.create(sample_dir)
n_meth <- round(msites$score * msites$coverage)
write.table(
    data.frame(
        chr = as.character(GenomicRanges::seqnames(msites)),
        start = GenomicRanges::start(msites),
        end = GenomicRanges::end(msites),
        percent = msites$score * 100,
        meth = n_meth,
        unmeth = msites$coverage - n_meth
    ),
    file.path(sample_dir, "sample_1.cov"),
    sep = "\t", row.names = FALSE, col.names = FALSE, quote = FALSE
)
write.table(
    data.frame(sampleName = "sample_1", bedFile = "sample_1.cov"),
    file.path(sample_dir, "samples.tsv"),
    sep = "\t", row.names = FALSE, quote = FALSE
)

devs <- runMethylTFR(
    sample_ann = "samples.tsv",
    sample_dir = sample_dir,
    tf_bindsites = tf_bindsites,
    gcfreqs = gcfreqs,
    gc_dist = gcdist,
    filetype = "bismarkCov"
)
deviations(devs)

unlink(sample_dir, recursive = TRUE)
```

---

runMethylTFRFromRnBeads

runMethylTFRFromRnBeads

Description

Run the methylTFR workflow directly on a preprocessed **RnBeads** object, without exporting per-sample BED files first.

Usage

```
runMethylTFRFromRnBeads(
  rnb_set,
  tf_bindsites = NULL,
  gcfreqs = NULL,
  gc_dist = NULL,
  chunkSize = 20,
  threads = 1,
  enhancer = NULL,
  ignoreStrand = TRUE,
  cov_threshold = 1,
  dpval_threshold = 0.05,
  sample_ann = NULL,
  tempDir = file.path(tempdir(), "methylTFR")
)
```

Arguments

`rnb_set`	A preprocessed `RnBSet` object, for example loaded with `RnBeads::load.rnb.set()`.
`tf_bindsites`	A `GRangesList` named by motif. Each element is a `GRanges` of the occurrences of that motif. The names must include every name in `names(gcfreqs)`. Get it with `getTFbindsites(motifSet = "jaspar2020")` from the **methylTFRAnnotationHg38** (or **methylTFRAnnotationMm10**) package, or build a custom set with **methylTFRAnnotationBuilder**.
`gcfreqs`	A named `list` with one element per motif, named as in `tf_bindsites`. Each element is a numeric matrix with the five GC bins in rows and the positions across the binding-site window in columns. Get it with `getGCfreq(motifSet = ...)` from the annotation package.
`gc_dist`	A `GRanges` object of 30-bp genomic windows with an integer metadata column `GC_bin` (1 to 5). Get it with `getGenomeGC()` from the annotation package.
`chunkSize`	Number of motifs processed together in one block. Default 20.
`threads`	Number of workers used to process the motifs of a chunk in parallel. Default 1 (serial).
`enhancer`	An optional `GRanges` object of regions, for example distal regulatory elements. Binding sites outside these regions are dropped. Any region set imported as `GRanges` works, for example the ENCODE candidate cis-regulatory elements (SCREEN) read with `rtracklayer::import()`. Default `NULL` uses all binding sites.
`ignoreStrand`	Logical. If `TRUE` (default), strand is ignored when overlapping methylation sites with annotations.

`cov_threshold` Numeric, minimum coverage of a retained site. Used only for sequencing sets. Default 1.

`dpval_threshold` Numeric, maximum detection p-value of a retained probe. Used only for array sets. Default 0.05.

`sample_ann` Optional `data.frame` with one row per sample, in the same order as the samples in `rnb_set`. All columns are kept as `colData` of the result; no column is required. Default `NULL` uses `RnBeads::pheno(rnb_set)`.

`tempDir` Directory for the temporary HDF5 files written during a run. Default is a folder under `tempdir()`. The files are deleted when the run finishes.

Details

Methylation calls are read at single-cytosine resolution. Sequencing sets (`RnBiseqSet`) are filtered by coverage, array sets (`RnBeadSet`) by detection p-value. The number of sites retained per sample is reported through **logger**.

Value

A `methylTFRdeviations` object with the assays `deviations`, `z` and `expected`, motifs in rows and samples in columns.

See Also

`runMethylTFR` for running methylTFR from per-sample BED files.

Examples

```
# A minimal end-to-end run on the BATF example data bundled with the
# package. The bundled calls are wrapped in an RnBiseqSet so that the
# example exercises the same code path as a preprocessed RnBeads object.
load(system.file("extdata", "example_data.rda", package = "methylTFR"))
load(system.file("extdata", "BATF_tf_bindsites.rda", package = "methylTFR"))
load(system.file("extdata", "BATF_gcfreqs.rda", package = "methylTFR"))
load(system.file("extdata", "gcdist_subset.rda", package = "methylTFR"))

if (requireNamespace("RnBeads", quietly = TRUE) &&
    requireNamespace("RnBeads.hg38", quietly = TRUE)) {
    # identifiers.column is a session option, not a property of the object,
    # so it has to be set before the set is built or samples() falls back
    # to row numbers and methylTFR cannot name the columns it returns
    old_ids <- RnBeads::rnb.getOption("identifiers.column")
    RnBeads::rnb.options(identifiers.column = "sampleName")

    sites <- data.frame(
        chr = as.character(GenomicRanges::seqnames(msites)),
        start = GenomicRanges::start(msites),
        strand = "*",
        stringsAsFactors = FALSE
    )
    rnb_set <- RnBeads::RnBiseqSet(
```

```
        pheno = data.frame(sampleName = "sample_1"),
        sites = sites,
        meth = matrix(msites$score, ncol = 1,
                      dimnames = list(NULL, "sample_1")),
        covg = matrix(msites$coverage, ncol = 1,
                      dimnames = list(NULL, "sample_1")),
        assembly = "hg38",
        summarize.regions = FALSE
    )

    devs <- runMethylTFRFromRnBeads(
        rnb_set = rnb_set,
        tf_bindsites = tf_bindsites,
        gcfreqs = gcfreqs,
        gc_dist = gcdist
    )
    RnBeads::rnb.options(identifiers.column = old_ids)
    deviations(devs)
}
```

Index

deviationZScores, *12*

addGCBinToMethylome, 2, *5*, *16*, *18*

cbind, *16*
cbind,methylTFRdeviations-method, 3
computeDeviation, *3*, 4, *13*, *16*, *18*
computeZScoreVariability, 6, *11*

deviations, 8, *12*, *18*
deviations,methylTFRdeviations-method, 8
deviationZScores, *7*, 9, *18*
deviationZScores,methylTFRdeviations-method, 10
differentialDeviationTest, *7*, 10

methylTFRdeviations, *11*, *18*, *21*
methylTFRdeviations-class, 12

p.adjust, *7*, *11*
plotExpectedFootprint, 12, *15*
plotMotifFootprint, *6*, *13*, 14

rbind,methylTFRdeviations-method, 15
readMethylome, *3*, *5*, *13*, *14*, 16, *18*
runMethylTFR, *6*, *12*, *16*, 17, *21*
runMethylTFRFromRnBeads, *12*, *18*, 19

SummarizedExperiment, *12*