

Package: methylTFR (via r-universe)

September 29, 2026

Title Quantification of DNA Methylation Signatures in TFBS

Version 0.99.9

Date 2026-09-29

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, RefManageR, RnBeads, RnBeads.hg38, rmarkdown, sessioninfo, 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://biocstaging.r-universe.dev
Date/Publication 2026-09-29 14:47:24 UTC
RemoteUrl https://github.com/BiocStaging/methylTFR
RemoteRef HEAD
RemoteSha 22b85e97f96ec40c9110ce35c73c1402542404ff

Contents

addGCBintoMethylome	2
cbind,methylTFRdeviations-method	3
computeDeviation	4
computeZScoreVariability	5
deviations	7
deviations,methylTFRdeviations-method	8
deviationZScores	8
deviationZScores,methylTFRdeviations-method	9
differential_deviation_test	9
methylTFRdeviations-class	11
plotExpectedFootprint	11
plotMotifFootprint	13
rbind,methylTFRdeviations-method	14
read_methylome	15
run_methyltfr	16
run_methylTFR_RnBeads	18
Index	20

addGCBintoMethylome	<i>addGCBintoMethylome</i>
---------------------	----------------------------

Description

This function is used to add GC bin to the methylation sites and calculate the average methylation for each bin.

Usage

addGCBintoMethylome(msites, gcdist, ignoreStrand = TRUE)

Arguments

- | | |
|--------------|---|
| msites | Imported methylation sites using read_methylome function |
| gcdist | a GRanges object contains Genome wide GC distribution |
| ignoreStrand | <ul style="list-style-type: none">if TRUE, it ignores strand info from annotation |

Value

a matrix object with GC bin with corresponding avg methylation

Author(s)

Irem Gunduz

Examples

```
library(methylTFR)

# Load the data
load(system.file("extdata",
  "gcdist_subset.rda",
  package = "methylTFR"
))
load(system.file("extdata",
  "example_data.rda",
  package = "methylTFR"
))

# Add GC bin
bin_meth <- addGCBintoMethylome(msites, gcdist)
```

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	<i>computeDeviation</i>
------------------	-------------------------

Description

computeDeviation is a function to calculate the deviation in transcription factor binding sites for a given motif

Usage

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

Arguments

motif	A character vector containing motif name
msites	imported methylation sites
tf_bindsites	a GRangesList object contains tf binding sites positions
gcfreqs	a list of GC bin frequency tables (matrices for multiple motif)
enhancer	a GRanges object specifying regions such as distal motif (optional)
ignoreStrand	if TRUE, it ignores strand info from annotation
binMsites	A matrix object with GC bin with corresponding avg methylation

Value

a numeric deviation score for a given motif

Examples

```
library(methylTFR)

# Load the data
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"
))

# Compute binMsites
bin_meth <- addGCBintoMethylome(msites, gcdist, TRUE)

# Compute the deviation
devs <- computeDeviation("BATF",
  msites,
  tf_bindsites,
  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

[differential_deviation_test](#) for testing activity differences between predefined groups.

Examples

```
# Load example data
load(system.file("extdata", "tc_mem.rda", package = "methyITFR"))
load(system.file("extdata", "tc_naive.rda", package = "methyITFR"))
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 methyITFRdeviations object.

Usage

```
deviations(x)
```

Arguments

x A methyITFRdeviations object.

Value

A matrix of deviations.

Examples

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

```
differential_deviation_test
differential_deviation_test
```

Description

Differential analysis is to test which motifs are having significant deviations among different groups.
Adapted from chromVAR::differential_test

Usage

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

Arguments

<code>deviations</code>	a data.frame contains deviations for each motif
<code>groups</code>	a character vector of group names or <code>colnames(deviations)</code>
<code>motifs</code>	a character vector of motif names used in analysis or <code>rownames(deviations)</code>
<code>alternative</code>	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
<code>parametric</code>	logical, if TRUE, parametric tests are used, otherwise non-parametric tests are used.
<code>padjMethod</code>	method for p-value adjustment, default is "BH"

Value

a data.frame contains motifs with corresponding p-value and adjusted p-value

Examples

```
# Load example data
load(system.file("extdata",
  "tc_mem.rda",
  package = "methy1TFR"
))
load(system.file("extdata",
  "tc_naive.rda",
  package = "methy1TFR"
))
devs <- cbind(tc_mem, tc_naive)

# Construct group labels from sample names
get_groupname <- function(x) {
  return(unlist(strsplit(x, split = "_"))[1])
}
groups <- sub(".bedGraph", "", unlist(lapply(
  FUN = get_groupname,
  X = colnames(devs)
)))

# Perform differential analysis
tc_result <- differential_deviation_test(
```

```

    deviations = devs,
    groups = groups,
    motifs = rownames(devs),
    alternative = "two.sided",
    parametric = TRUE,
    padjMethod = "BH"
)

```

```

methyITFRdeviations-class
      methyITFRdeviations

```

Description

Class for storing results from `run_methyITFR` function.

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

Creates a footprint plot of expected vs observed methylation for a given motif and a single methylation site.

Usage

```

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

```

Arguments

<code>motif</code>	Motif name as a character string
<code>tf_bindsites</code>	Transcript Factor binding sites from the annotation package
<code>msites</code>	Methylation sites as a data frame
<code>sample_name</code>	Optional sample name as a character string, defaults to the file name if not provided
<code>gc_dist</code>	a GRanges object contains Genome wide GC distribution
<code>gcfreqs</code>	GC frequency of the genome used to compute expected methylation
<code>enhancer</code>	Specific region such as distal motif, proximal motif
<code>returnPlotData</code>	Logical indicating whether to return the plot data

Value

A ggplot object of TF footprint plot for a single sample

Examples

```
library(ggplot2)
library(methylTFR)

# Load the data
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"
))

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

plotMotifFootprint	<i>plotMotifFootprint</i>
--------------------	---------------------------

Description

Creates a footprint plot of bias corrected methylation for a given motif and sample

Usage

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

Arguments

motif	Motif name as a character string
tf_bindsites	Transcript Factor binding sites from the annotation package
msites	Methylation sites as a data frame
sample_name	Optional sample name as a character string, defaults to the file name if not provided
gc_dist	a GRanges object contains Genome wide GC distribution
gcfreqs	GC frequency of the genome used to compute expected methylation
enhancer	Specific region such as distal motif, proximal motif
method	Method to calculate the difference, either "subtraction" or "division"
flankNorm	Numeric value indicating the flank normalization distance, default is 50

Value

A ggplot object of TF footprint difference plot for a single sample

Examples

```
library(ggplot2)  
library(methylTFR)  
  
# Load the data  
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"
  ))

  # Plot the expected footprint
  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, ])
```

read_methylome	<i>read_methylome</i>
----------------	-----------------------

Description

read_methylome is a function to import methylation data into a GRanges object. Bed file should be in EPP, ALLC or BisSNP format.

Usage

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

Arguments

filename	filename which contains methylation data
type	Type of file format. Currently supported epp, bisnp, bismarkCytosine, bismark-cov, allc and encode
cov_threshold	numeric, coverage threshold to filter out low coverage sites, default is 1

Value

a GenomicRange object with methylation, coverage information

Author(s)

Irem Gunduz

Examples

```
# Read bisnp file
bisnp_path <- system.file(
  "extdata",
  "bisnp.tsv.gz",
  package = "methylTFR"
)
bisnp <- read_methylome(bisnp_path, "bisnp")
```

run_methyltf	<i>run_methyltf</i>
--------------	---------------------

Description

This function is a wrapper function to calculate the deviation in transcription factor footprint base for all given motifs per raw samples

Usage

```
run_methyltf(
  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
)
```

Arguments

sample_ann	A tab seperated file contains sample annotations
sample_dir	The directory where all bed file and annotation file stored
tf_bindsites	a GRangesList object contains tf binding sites positions
gcfreqs	a list of GC bin frequency tables (matrices for multiple motif)
gc_dist	a GRanges object contains Genome wide GC distribution
sampleColName	column name of the sample bed file in the annotation file
chunkSize	Chunk size for parallel processing of motifs (default: 20)
full_path	if TRUE, the bed file path in the annotation file is full path
annfile	if provided, the sample annotation file is not read from the sample_dir
threads	Thread count for parallel processing
enhancer	a GRanges object specifying regions such as distal motif (optional)
filetype	file type of the bed file, currently supported: bismark, bismarkcytosine, bismarkcov, encode
ignoreStrand	if TRUE, it ignores strand info from annotation
cov_threshold	numeric, coverage threshold to filter out low coverage sites, default is 1

Value

a methylTFRdeviations object with bias-corrected deviation and Z-scores

See Also

[run_methylTFR_RnBeads](#) for running methylTFR directly on a preprocessed RnBeads object.

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"))

# run_methyltfr() 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 <- run_methyltfr(
  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)
```

```
run_methylTFR_RnBeads  run_methylTFR_RnBeads
```

Description

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

Usage

```
run_methylTFR_RnBeads(
  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
)
```

Arguments

rnb_set	A preprocessed RnBSet object.
tf_bindsites	a GRangesList of TF binding site positions.
gcfreqs	a list of GC bin frequency tables.
gc_dist	a GRanges of the genome-wide GC distribution.
chunkSize	Chunk size for parallel processing of motifs.
threads	Thread count for parallel processing.
enhancer	an optional GRanges of regions to restrict to.
ignoreStrand	if TRUE, strand information is ignored.
cov_threshold	numeric, minimum coverage of a retained site.
dpval_threshold	numeric, maximum detection p-value of a retained probe.
sample_ann	Optional data.frame of sample annotation.

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 bias-corrected deviations and Z-scores.

See Also

[run_methyltfr](#) 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 <- run_methylTFR_RnBeads(
    rnb_set = rnb_set,
    tf_bindsites = tf_bindsites,
    gcfreqs = gcfreqs,
    gc_dist = gcdist
  )
  RnBeads::rnb.options(identifiers.column = old_ids)
  deviations(devs)
}
```

Index

deviationZScores, [11](#)

addGCBintoMethylome, [2](#)

cbind, [14](#)

cbind,methylTFRdeviations-method, [3](#)

computeDeviation, [4](#)

computeZScoreVariability, [5](#)

deviations, [7](#), [11](#)

deviations,methylTFRdeviations-method,
[8](#)

deviationZScores, [6](#), [8](#)

deviationZScores,methylTFRdeviations-method,
[9](#)

differential_deviation_test, [7](#), [9](#)

methylTFRdeviations-class, [11](#)

p.adjust, [6](#)

plotExpectedFootprint, [11](#)

plotMotifFootprint, [13](#)

rbind,methylTFRdeviations-method, [14](#)

read_methylome, [15](#)

run_methyltfr, [11](#), [16](#), [19](#)

run_methylTFR_RnBeads, [17](#), [18](#)

SummarizedExperiment, [11](#)