Package 'methylTFR'

November 25, 2026

Title Quantification of DNA Methylation Signatures in TFBS

Version 1.0.0

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

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

git_branch RELEASE_3_24

git_last_commit 84c3037

git_last_commit_date 2026-11-23

Repository Bioconductor 3.24

Date/Publication 2026-11-24

Author Irem B. Gündüz [aut, cre] (ORCID:
<https://orcid.org/0000-0003-2641-0916>),
Sarath Kumar Murugan [aut] (ORCID:
<https://orcid.org/0000-0001-7378-7869>),
Fabian Mueller [aut] (ORCID: <https://orcid.org/0000-0001-5809-2321>),
ERA-NET Transcan-Neu III - EPILUNAR [fnd] (Grant 01KT2409),
Saarland University [fnd] (NanoBioMed Young Investigator Grant)

Maintainer Irem B. Gündüz <irembgunduz@gmail.com>

Contents

methylTFR-package . . . 4
addGCBinToMethylome . . . 5
anova_helper . . . 6
assemble_core_result . . . 6
bootstrap_variability . . . 7
bpparam_from_threads . . . 7
calibrateDeviations . . . 8
cbind,methylTFRdeviations-method . . . 8
check_annotation_inputs . . . 9
check_core_inputs . . . 10
check_deviation_inputs . . . 10
check_enhancer . . . 11
check_file_run_inputs . . . 12
check_footprint_inputs . . . 12
check_gcfreqs . . . 13
check_gc_dist . . . 13
check_msites . . . 14
check_rnb_inputs . . . 14
check_run_options . . . 15
check_tf_bindsites . . . 15
check_variability_inputs . . . 16
computeDeviation . . . 16
computeExpectations . . . 18
computeExpectedFootprint . . . 18
computeFootprint . . . 19
computeRowZScore . . . 20
computeZScoreVariability . . . 20
create_sink . . . 22
deviations . . . 23
deviations,methylTFRdeviations-method . . . 23
deviationZScores . . . 24
deviationZScores,methylTFRdeviations-method . . . 24
dev_helper . . . 25
differentialDeviationTest . . . 25
diff_pvalues . . . 26
expected_footprint_plot . . . 27

footprint_difference . . . 27
granges_helper . . . 28
group_mean_difference . . . 28
kw_helper . . . 29
locate_sample_files . . . 29
methylTFRdeviations-class . . . 30
methyltfr_core . . . 30
normalise_footprint_flank . . . 31
parse_allc . . . 32
parse_bismarkcov . . . 32
parse_bismarkcytosine . . . 33
parse_bissnp . . . 33
parse_encode . . . 34
parse_epp . . . 34
plotExpectedFootprint . . . 35
plotMotifFootprint . . . 36
process_core_sample . . . 38
rbind,methylTFRdeviations-method . . . 39
readMethylome . . . 40
read_sample_annotation . . . 41
resolve_annotation_file . . . 41
resolve_diff_groups . . . 42
resolve_rnb_sample_ann . . . 42
rnb_annotation_by_id . . . 43
rnb_annotation_table . . . 43
rnb_annotation_target . . . 44
rnb_column . . . 44
rnb_has_coverage . . . 45
rnb_has_dpval . . . 45
rnb_probe_ids . . . 46
rnb_quality_mode . . . 46
rnb_sample_ids . . . 47
rnb_sample_msites . . . 47
rnb_sites_to_granges . . . 48
runMethylTFR . . . 48
runMethylTFRFromRnBeads . . . 50
set_grid . . . 53
t_helper . . . 53
valid_core_motifs . . . 54
variability_test . . . 54
wilcoxon_helper . . . 55
write_block_to_sink . . . 55

Index **56**

methylTFR-package *methylTFR: transcription factor activity from DNA methylation*

Description

Scores the methylation footprint at transcription factor binding sites and returns one bias-corrected score per motif and sample.

Getting started

1. Import methylation calls with `readMethylome()`.
2. Load annotations from methylTFRAnnotationHg38 (`getTFbindsites()`, `getGCfreq()`, `getGenomeGC()`).
3. Compute scores with `runMethylTFR()` or `runMethylTFRFromRnBeads()`.
4. Explore with `deviations()`, `deviationZScores()`, `computeZScoreVariability()`, `differentialDeviationT` `plotMotifFootprint()` and `plotExpectedFootprint()`.

See `vignette("methylTFR")` for a worked example.

Bundled data

The package code is MIT licensed (see `LICENSE`). The files in `inst/extdata` have the following sources and terms.

- `example_data.rda`, `msites_sub.rda`, `gcdist_subset.rda`, `gcdist_BATF.rda`: CpG methylation calls, and the GC windows that overlap them, from ENCODE whole-genome bisulfite sequencing experiment ENCSR663MXB (GRCh38, https://www.encodeproject.org/experiments/ENCSR663MXB/). ENCODE data are released without restrictions on use (https://www.encodeproject.org/help/citing-encode/). Please acknowledge the ENCODE Consortium and the production laboratory, and cite ENCODE Project Consortium (2012) Nature 489:57-74 (doi:10.1038/nature11247) and Luo et al. (2020) Nucleic Acids Res 48:D882-D889 (doi:10.1093/nar/gkz1062).
- `BATF_tf_bindsites.rda`, `BATF_gcfreqs.rda`: BATF binding sites on chr1 and the genome-wide GC frequencies of the motif, derived from the JASPAR2020 BATF motif through methylTFRAnnotationHg38. JASPAR data are released under CC BY 4.0 (https://jaspar.elixir.no/about/). Cite Fornes et al. (2020) Nucleic Acids Res 48:D87-D92 (doi:10.1093/nar/gkz1001).
- `immuneDeviations.rda`, `tc_mem.rda`, `tc_naive.rda`: simulated deviation scores. The source pseudobulk methylomes (Gunduz et al. 2025, bioRxiv, doi:10.1101/2025.09.09.675101; Wang et al. 2025, Nat Genet, doi:10.1038/s41588-025-02479-6) are under controlled access in dbGaP (phs003204.v1.p1). Only per-cell-type motif means and SDs were taken from them; every sample was drawn from a normal distribution with those parameters. No individual-level values, methylation calls or donor identifiers are included. Released under the package's MIT license. Made by `inst/scripts/tcell_data.R` and `inst/scripts/vignette_data.R`.
- `*.tsv.gz`: small simulated files in each supported input format, made by `inst/scripts/read_methylome_data.R`

Author(s)

Maintainer: Irem B. Gündüz <irembgunduz@gmail.com> (ORCID)

Authors:

- Sarath Kumar Murugan <sarath.murugan@outlook.com> (ORCID)

- Fabian Mueller <fabian.mueller@uni-saarland.de> (ORCID)

Other contributors:

- ERA-NET Transcan-Neu III - EPILUNAR (Grant 01KT2409) [funder]
- Saarland University (NanoBioMed Young Investigator Grant) [funder]

See Also

Useful links:

- https://github.com/EpigenomeInformatics/methylTFR
- https://epigenomeinformatics.github.io/methylTFR/
- Report bugs at https://support.bioconductor.org/t/methylTFR

`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
```

---

anova_helper *anova test for multiple groups*

Description

anova test for multiple groups

Usage

```
anova_helper(x, groups)
```

Arguments

x	a numeric vector of deviations
groups	a character vector of group names

Value

a p-value

---

assemble_core_result *assemble_core_result*

Description

Close the sinks and assemble the deviations, their row-wise Z-scores and the expected deviations into a result object.

Usage

```
assemble_core_result(dev_sink, exp_sink, samples)
```

Arguments

dev_sink	The sink holding the bias-corrected deviations.
exp_sink	The sink holding the expected deviations.
samples	A data.frame with one row per sample.

Value

a methylTFRdeviations object.

bootstrap_variability *bootstrap_variability*

Description

Add bootstrap confidence bounds to a variability table.

Usage

```
bootstrap_variability(z, res, niterations, conf_level)
```

Arguments

z	The calibrated Z-score matrix.
res	The variability table to add the bounds to.
niterations	Number of bootstrap iterations.
conf_level	Confidence level of the bounds.

Value

`res` with the two bound columns added.

bpparam_from_threads *bpparam_from_threads*

Description

Build the **BiocParallel** back-end used to spread the motifs of one chunk over workers.

Usage

```
bpparam_from_threads(threads)
```

Arguments

threads	Thread count for parallel processing.

Details

A forking back-end is used where the platform supports it and a socket back-end on Windows, so that `threads` has the same meaning on every platform. `threads = 1` runs serially in the current process.

Value

A `BiocParallelParam` object.

calibrateDeviations *calibrateDeviations*

Description

Standardise bias-corrected deviation scores against a within-sample null, so that the resulting scores are comparable across motifs and approximately standard normal in the absence of TF-specific signal.

Usage

```
calibrateDeviations(devs, method = c("robust", "gaussian"))
```

Arguments

devs	A numeric matrix 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).

Details

Each column (sample) is centred and scaled independently, using the distribution of deviation scores across all motifs in that sample as the null. With several hundred motifs, the great majority of which carry no strong methylation signal in any single sample, this distribution is a usable estimate of the sample's noise level.

This is deliberately different from computeRowZScore, which standardises each motif against its own spread across samples. Row-wise Z-scores are the right transform for visualising a heatmap, but they force the standard deviation of every row to one and therefore cannot be used to rank motifs by variability.

Value

A numeric matrix of the same dimensions as devs.

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)
```

---

check_annotation_inputs

check_annotation_inputs

Description

Validate the annotation objects required by both methylTFR entry points.

Usage

```
check_annotation_inputs(tf_bindsites, gcfreqs, gc_dist, enhancer = NULL)
```

Arguments

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.

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.

Value

invisible TRUE, called for the side effect of signalling errors.

`check_core_inputs` *check_core_inputs*

Description

Validate the arguments shared by both methylTFR entry points.

Usage

```
check_core_inputs(sample_ids, msites_fun, samples, tempDir)
```

Arguments

sample_ids	A character vector of sample identifiers.
msites_fun	A function of a single integer sample index.
samples	A data.frame with one row per sample.
tempDir	Directory for the temporary HDF5 files.

Value

Invisible NULL. Called for the errors it raises.

`check_deviation_inputs` *check_deviation_inputs*

Description

Validate the inputs of computeDeviation.

Usage

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

Arguments

motif	Motif name as a character string.
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.
binMsites	The matrix returned by addGCBinToMethylome, with the columns gcbin and avg_mscore.
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.
extendBy	Number of base pairs added to each binding site.
windowBreaks	Break points of the distance bins.

Value

Invisible NULL. Called for the errors it raises.

---

check_enhancer *check_enhancer*

Description

Check the optional region set.

Usage

```
check_enhancer(enhancer)
```

Arguments

enhancer	The object to check, or NULL.

Value

Invisible NULL. Called for the errors it raises.

check_file_run_inputs *check_file_run_inputs*

Description

Validate the file-specific arguments of `runMethylTFR`.

Usage

```
check_file_run_inputs(filetype, sampleColName, full_path)
```

Arguments

`filetype`	File type of the methylation call files.
`sampleColName`	Column name holding the file names.
`full_path`	if TRUE, the annotation file holds full paths.

Value

Invisible `NULL`. Called for the errors it raises.

check_footprint_inputs *check_footprint_inputs*

Description

Validate the inputs shared by the two footprint plotting functions.

Usage

```
check_footprint_inputs(
  motif,
  tf_bindsites,
  msites,
  gc_dist,
  gcfreqs,
  enhancer = NULL
)
```

Arguments

`motif`	Motif name as a character string.
`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.
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.

Value

Invisible NULL. Called for the errors it raises.

---

check_gcfreqs *check_gcfreqs*

Description

Check that GC frequency tables are a named list.

Usage

```
check_gcfreqs(gcfreqs)
```

Arguments

gcfreqs	The object to check.

Value

Invisible NULL. Called for the errors it raises.

---

check_gc_dist *check_gc_dist*

Description

Check that the genome-wide GC distribution is a GRanges object with a GC_bin column.

Usage

```
check_gc_dist(gc_dist)
```

Arguments

gc_dist	The object to check.

Value

Invisible NULL. Called for the errors it raises.

---

check_msites *check_msites*

Description

Check that methylation sites have the structure returned by readMethylome.

Usage

```
check_msites(msites)
```

Arguments

msites	The object to check.

Value

Invisible NULL. Called for the errors it raises.

---

check_rnb_inputs *check_rnb_inputs*

Description

Check that **RnBeads** is available and that the object handed in is an RnBSet.

Usage

```
check_rnb_inputs(rnb_set)
```

Arguments

rnb_set	The object to validate.

Value

Invisible NULL. Called for the errors it raises.

check_run_options *check_run_options*

Description

Validate the run-time options shared by both methylTFR entry points, substituting documented defaults where a value is unusable.

Usage

```
check_run_options(
  chunkSize = 20,
  threads = 1,
  ignoreStrand = TRUE,
  cov_threshold = 1
)
```

Arguments

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).
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.

Value

A named list with the validated values.

check_tf_bindsites *check_tf_bindsites*

Description

Check that TF binding sites are a named list of ranges.

Usage

```
check_tf_bindsites(tf_bindsites)
```

Arguments

tf_bindsites	The object to check.

Value

Invisible NULL. Called for the errors it raises.

check_variability_inputs *check_variability_inputs*

Description

Validate the inputs of `computeZScoreVariability` and return the deviation scores as a matrix.

Usage

```
check_variability_inputs(object, bootstrap, conf_level)
```

Arguments

`object`	A `methylTFRdeviations` object, matrix or data.frame.
`bootstrap`	Logical, whether bootstrap bounds were requested.
`conf_level`	Confidence level of the bootstrap bounds.

Value

The deviation scores as a numeric matrix.

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
)
```

---

computeExpectations *computeExpectations*

Description

Expected methylation profile of a motif in one sample.

Usage

```
computeExpectations(binMsites, gcfreq)
```

Arguments

binMsites	The matrix returned by addGCBinToMethylome.
gcfreq	The GC bin frequency matrix of one motif.

Value

a data.table with the columns x (distance to the site centre) and avg_methyl.

---

computeExpectedFootprint *computeExpectedFootprint*

Description

Methylation profile expected from GC content around the binding sites of one motif.

Usage

```
computeExpectedFootprint(motif, gcfreqs, gc_dist, enhancer = NULL, msites)
```

Arguments

motif	Name of the motif, one of `names(tf_bindsites)`.
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.
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.
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`.

Value

a `data.table` with the columns `x` and `avg_methyl`.

`computeFootprint` *computeFootprint*

Description

Observed methylation profile around the binding sites of one motif.

Usage

```
computeFootprint(
  motif_name,
  tf_bindsites,
  msites,
  enhancer = NULL,
  extendBy = 130
)
```

Arguments

motif_name	Name of the motif.
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`.

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.
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.

Value

a data.table with one row per distance to the site centre.

---

computeRowZScore *Row-wise Z-score of a matrix*

Description

This function takes a matrix and computes row-wise z-scores.

Usage

```
computeRowZScore(mat)
```

Arguments

mat	A matrix

Value

A matrix with row-wise z-scores

---

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), ])
```

`create_sink` *create_sink*

Description

Create a temporary HDF5 sink for storing methylTFR results.

Usage

```
create_sink(
  files_list,
  motifs,
  tempDir = file.path(tempdir(), "methylTFR"),
  pattern = "methylTFR",
  fileext = ".h5",
  verbose = TRUE
)
```

Arguments

`files_list`	A character vector of sample names.
`motifs`	A character vector of motifs.
`tempDir`	Directory the HDF5 file is written to. Created if it does not exist.
`pattern`	Prefix of the temporary file name.
`fileext`	File extension of the temporary file.
`verbose`	A logical indicating whether to print messages.

Value

An `HDF5RealizationSink`. Its file is in `sink@filepath`.

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)
```

dev_helper *dev_helper*

Description

Ratio of the mean methylation in the centre bin to the mean of the two outer flank bins.

Usage

```
dev_helper(data, breaks = c(-250, -200, -25, 25, 200, 250))
```

Arguments

data	A data.table with the columns x (distance to the site centre) and avg_methyl.
breaks	Six strictly increasing break points defining five bins.

Value

A numeric value, or NA if the centre bin or either outer flank bin is empty.

---

differentialDeviationTest *differentialDeviationTest*

Description

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

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)
```

`diff_pvalues` *diff_pvalues*

Description

Test every motif for a difference between the groups, with the test chosen from the number of groups and `parametric`.

Usage

```
diff_pvalues(deviations, groups, parametric, alternative)
```

Arguments

`deviations`	A matrix of deviation scores, motifs in rows.
`groups`	The group labels as a `factor`.
`parametric`	if TRUE, use a t-test or ANOVA, otherwise a Wilcoxon or Kruskal-Wallis test.
`alternative`	The alternative hypothesis of the two-group tests.

Value

A numeric vector of p-values, one per motif.

`expected_footprint_plot` *expected_footprint_plot*

Description

Draw the observed and expected methylation profiles.

Usage

```
expected_footprint_plot(combined_data, motif, sample_name)
```

Arguments

`combined_data`	A `data.table` with the columns `x`, `avg_methyl` and `type`.
`motif`	Motif name as a character string.
`sample_name`	Sample label used in the title.

Value

A `ggplot` object.

`footprint_difference` *footprint_difference*

Description

Combine the observed and expected profiles into a single corrected curve.

Usage

```
footprint_difference(plot_data, method, sample_name)
```

Arguments

`plot_data`	A `data.table` with the columns `x`, `avg_methyl` and `type`.
`method`	Either `"substraction"` or `"division"`.
`sample_name`	Sample label used in the curve label.

Value

A `list` with the corrected `data` and the axis `lab`.

granges_helper *granges_helper*

Description

helper function to convert msites in EPP GRanges format

Usage

```
granges_helper(grobj, chr, mscore, cov, startP, endP)
```

Arguments

grobj	GRanges object
chr	chromosome
mscore	methylation score
cov	methylation coverage
startP	start position
endP	end position

Value

a GenomicRanges object with EPP format

group_mean_difference *group_mean_difference*

Description

Effect size per motif. For two groups, the mean of the first level minus the mean of the second level (signed). For more groups, the range of the group means (unsigned).

Usage

```
group_mean_difference(deviations, groups)
```

Arguments

deviations	A matrix of deviation scores, motifs in rows.
groups	The group labels as a factor.

Value

A numeric vector with one value per motif.

kw_helper *Kruskal-Wallis rank sum test for multiple groups*

Description

Kruskal-Wallis rank sum test for multiple groups

Usage

```
kw_helper(x, groups)
```

Arguments

x	a numeric vector of deviations
groups	a character vector of group names

Value

a p-value

locate_sample_files *locate_sample_files*

Description

Build and check the list of per-sample methylation files.

Usage

```
locate_sample_files(samples, sample_dir, sampleColName, full_path)
```

Arguments

samples	The sample annotation as a data.frame.
sample_dir	Directory holding the methylation call files.
sampleColName	Column name holding the file names.
full_path	if TRUE, the annotation file holds full paths.

Value

A character vector of existing file paths.

`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`).

`methyltfr_core` *methyltfr_core*

Description

Internal engine shared by `runMethylTFR` and `runMethylTFRFromRnBeads`. It validates the motif set, allocates the on-disk sinks, iterates over samples, computes per-motif deviations in chunks and assembles the resulting `methylTFRdeviations` object.

The only difference between the two public entry points is where the per-sample methylation calls come from. That difference is isolated in the `msites_fun` argument, so both entry points share identical numerical behaviour.

Usage

```
methyltfr_core(
  sample_ids,
  msites_fun,
  samples,
  tf_bindsites,
  gcfreqs,
  gc_dist,
  chunkSize = 20,
  threads = 1,
  enhancer = NULL,
  ignoreStrand = TRUE,
  tempDir = file.path(tempdir(), "methylTFR")
)
```

Arguments

`sample_ids`	A character vector of sample identifiers. Used for the column names of the resulting object and to size the sinks.
`msites_fun`	A function of a single integer `i` returning the methylation sites of sample `i` as a `GRanges` with the metadata columns `score` and `coverage`.

samples	A data.frame of sample annotation with one row per entry of sample_ids, used as colData.
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.
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.

Value

a methylTFRdeviations object with bias-corrected deviations, row-wise Z-scores and expected deviations.

---

normalise_footprint_flank

normalise_footprint_flank

Description

Normalise a corrected footprint against its outer flanking windows, on the same scale as the statistic itself.

Usage

```
normalise_footprint_flank(difference_data, method, flankNorm)
```

Arguments

difference_data	The corrected footprint as a data.table.
method	Either "substraction" or "division".
flankNorm	Width of the flanking window used for normalisation.

Details

Dividing a difference by its flank mean would rescale it by an arbitrary factor, because that mean is near zero: for a typical footprint it inflates the curve by an order of magnitude and can invert its sign.

Value

The normalised `data.table`.

---

parse_allc *Parse an ALLC file*

Description

Parse an ALLC file

Usage

```
parse_allc(filename)
```

Arguments

`filename`	Path to the file.

Value

A list with the elements msites, mscore and cov.

---

parse_bismarkcov *Parse a Bismark coverage file*

Description

Parse a Bismark coverage file

Usage

```
parse_bismarkcov(filename)
```

Arguments

`filename`	Path to the file.

Value

A list with the elements msites, mscore and cov.

parse_bismarkcytosine *Parse a Bismark cytosine report*

Description

Parse a Bismark cytosine report

Usage

```
parse_bismarkcytosine(filename)
```

Arguments

filename	Path to the file.

Value

A list with the elements msites, mscore and cov.

parse_bissnp *Parse a BisSNP file*

Description

Parse a BisSNP file

Usage

```
parse_bissnp(filename)
```

Arguments

filename	Path to the file.

Value

A list with the elements msites, mscore and cov.

parse_encode *Parse an ENCODE bedMethyl file*

Description

Parse an ENCODE bedMethyl file

Usage

```
parse_encode(filename)
```

Arguments

filename	Path to the file.

Value

A list with the elements msites, mscore and cov.

parse_epp *Parse an EPP file*

Description

Parse an EPP file

Usage

```
parse_epp(filename)
```

Arguments

filename	Path to the file.

Value

A list with the elements msites, mscore and cov.

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"
)
```

process_core_sample *process_core_sample*

Description

Compute and write the deviations of one sample, one motif chunk at a time.

Usage

```
process_core_sample(
  index,
  sample_ids,
  msites_fun,
  motif_chunks,
  tf_bindsites,
  gcfreqs,
  gc_dist,
  dev_grid,
  exp_grid,
  dev_sink,
  exp_sink,
  BPPARAM,
  enhancer,
  ignoreStrand
)
```

Arguments

index	Integer index of the sample within sample_ids.
sample_ids	A character vector of sample identifiers.
msites_fun	A function of a single integer sample index.
motif_chunks	A list of character vectors of motif names.
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.
dev_grid, exp_grid	The grids the blocks are written on.
dev_sink, exp_sink	The sinks the blocks are written to.
BPPARAM	A BiocParallelParam object.

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.

Value

Invisible NULL. Called for its effect on the sinks.

---

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")
```

`read_sample_annotation` *read_sample_annotation*

Description

Read a methylTFR sample annotation table.

Usage

```
read_sample_annotation(annfile, sampleColName)
```

Arguments

`annfile`	Path to a `.csv` or `.tsv` annotation file.
`sampleColName`	Name of the column holding the per-sample file names.

Value

A `data.frame` of sample annotation.

`resolve_annotation_file` *resolve_annotation_file*

Description

Work out the path of the sample annotation file.

Usage

```
resolve_annotation_file(annfile, sample_ann, sample_dir)
```

Arguments

`annfile`	Explicit path to the annotation file, or NULL.
`sample_ann`	Name of the annotation file inside `sample_dir`.
`sample_dir`	Directory holding the methylation call files.

Value

The path of the annotation file as a character string.

resolve_diff_groups *resolve_diff_groups*

Description

Derive and validate the group labels used by `differentialDeviationTest`.

Usage

```
resolve_diff_groups(deviations, groups)
```

Arguments

`deviations`	A matrix of deviation scores, motifs in rows.
`groups`	Group labels, or NULL to take them from the column names.

Value

The group labels as a `factor`.

resolve_rnb_sample_ann *resolve_rnb_sample_ann*

Description

Default the sample annotation to the phenotype table of the RnBeads object and check its shape.

Usage

```
resolve_rnb_sample_ann(rnb_set, sample_ann, sample_ids)
```

Arguments

`rnb_set`	A preprocessed `RnBSet` object.
`sample_ann`	Sample annotation supplied by the caller, or NULL.
`sample_ids`	Character vector of sample identifiers.

Value

The sample annotation as a `data.frame`.

rnb_annotation_by_id *rnb_annotation_by_id*

Description

Subset a genome-wide annotation track to the probes of an RnBeads object, matching on probe identifier.

Usage

```
rnb_annotation_by_id(rnb_set, target, assembly)
```

Arguments

rnb_set	An RnBSet object.
target	Character scalar naming the annotation target.
assembly	Character scalar naming the genome assembly.

Details

The sites slot holds a three-column index matrix, not a row index into the annotation, so the track is matched by identifier rather than by position.

Value

A data.frame of annotation rows, or NULL when the identifiers cannot be matched.

rnb_annotation_table *rnb_annotation_table*

Description

Look up the site or probe annotation of an RnBeads object.

Usage

```
rnb_annotation_table(rnb_set, target, assembly)
```

Arguments

rnb_set	An RnBSet object.
target	Character scalar naming the annotation target.
assembly	Character scalar naming the genome assembly.

Details

The annotation stored in the object is used when available, otherwise the genome-wide track registered for target is matched to the probes of the object by identifier.

Value

A data.frame with one row per site or probe.

rnb_annotation_target *rnb_annotation_target*

Description

Resolve the annotation target of an RnBeads object.

Usage

```
rnb_annotation_target(rnb_set)
```

Arguments

rnb_set	An RnBSet object.

Value

A character scalar naming the annotation target, "sites" for sequencing sets and the array platform for array sets.

rnb_column *rnb_column*

Description

Extract a single sample column from an RnBeads accessor.

Usage

```
rnb_column(accessor, rnb_set, index)
```

Arguments

accessor	An RnBeads accessor function.
rnb_set	An RnBSet object.
index	Integer index of the sample to extract.

Value

A numeric vector with one value per site.

rnb_has_coverage *rnb_has_coverage*

Description

Test whether an RnBeads object carries coverage information.

Usage

```
rnb_has_coverage(rnb_set)
```

Arguments

rnb_set	An RnBSet object.

Value

A logical scalar.

rnb_has_dpval *rnb_has_dpval*

Description

Test whether an RnBeads object carries detection p-values.

Usage

```
rnb_has_dpval(rnb_set)
```

Arguments

rnb_set	An RnBSet object.

Value

A logical scalar.

rnb_probe_ids *rnb_probe_ids*

Description

Read the probe identifiers of an RnBeads object.

Usage

```
rnb_probe_ids(rnb_set)
```

Arguments

rnb_set	An RnBSet object.

Details

The sites slot of an RnBSet is an index matrix whose row names carry the probe identifiers.

Value

A character vector of probe identifiers, or NULL.

rnb_quality_mode *rnb_quality_mode*

Description

Decide which per-site quality filter applies to an RnBeads object and report it.

Usage

```
rnb_quality_mode(rnb_set, dpval_threshold)
```

Arguments

rnb_set	An RnBSet object.
dpval_threshold	numeric detection p-value threshold.

Value

A list with the logical flags has_covg and has_dpval.

rnb_sample_ids *rnb_sample_ids*

Description

Determine the sample identifiers of an RnBeads object.

Usage

```
rnb_sample_ids(rnb_set)
```

Arguments

rnb_set	An RnBSet object.

Value

A character vector of sample identifiers.

rnb_sample_msites *rnb_sample_msites*

Description

Extract the methylation calls of a single sample from an RnBeads object as a GRanges object.

Usage

```
rnb_sample_msites(
  rnb_set,
  sites_gr,
  index,
  cov_threshold = 1,
  has_covg = TRUE,
  dpval_threshold = 0.05,
  has_dpval = FALSE
)
```

Arguments

rnb_set	An RnBSet object.
sites_gr	A GRanges object of site positions.
index	Integer index of the sample to extract.
cov_threshold	numeric coverage threshold.
has_covg	logical, whether coverage filtering applies.
dpval_threshold	numeric detection p-value threshold.
has_dpval	logical, whether detection p-value filtering applies.

Value

A GRanges object restricted to valid methylation calls.

`rnb_sites_to_granges` *rnb_sites_to_granges*

Description

Build a GRanges object of the site or probe annotation of an RnBeads object.

Usage

```
rnb_sites_to_granges(rnb_set, ignoreStrand = TRUE)
```

Arguments

rnb_set	An RnBSet object.
ignoreStrand	if TRUE, all ranges are returned with strand "*".

Value

A GRanges object with one range per site or probe.

`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)
}
```

`set_grid` *set_grid*

Description

Set the grid for the methylTFR sink

Usage

```
set_grid(files_list, motif_chunks)
```

Arguments

`files_list`	A character vector of file names
`motif_chunks`	A list of motif chunks

Value

A methylTFR grid

`t_helper` *t-test for two groups*

Description

Welch t-test for two groups

Usage

```
t_helper(x, groups, alternative)
```

Arguments

`x`	a numeric vector of deviations
`groups`	a character vector of group names
`alternative`	the alternative hypothesis, one of "two.sided", "greater" or "less".

Value

a p-value

valid_core_motifs *valid_core_motifs*

Description

Drop motifs whose binding sites are empty or whose GC bin frequency matrix is missing.

Usage

```
valid_core_motifs(tf_bindsites, gcfreqs)
```

Arguments

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.

Value

A character vector of the motif names that can be processed.

variability_test *variability_test*

Description

Summarise the calibrated Z-scores into a variability per motif and test each against the chi-squared null.

Usage

```
variability_test(z, motif_names, padjMethod)
```

Arguments

z	The calibrated Z-score matrix.
motif_names	Character vector of motif names.
padjMethod	Multiple testing correction passed to stats::p.adjust.

Value

A data.frame with one row per motif.

wilcoxon_helper *wilcoxon test for two groups*

Description

wilcoxon test for two groups

Usage

```
wilcoxon_helper(x, groups, alternative)
```

Arguments

x	a numeric vector of deviations
groups	a character vector of group names
alternative	the alternative hypothesis, one of "two.sided", "greater" or "less".

Value

a p-value

write_block_to_sink *write_block_to_sink*

Description

Write the block to the methylTFR sink

Usage

```
write_block_to_sink(dev_values, grid, i, j, sink)
```

Arguments

dev_values	A numeric vector of deviation values
grid	A methylTFR grid
i	An integer specifying the row index
j	An integer specifying the column index
sink	A methylTFR sink

Value

Invisible NULL. Called for its effect on the sink.

Index

∗ **internal**
anova_helper, 6
assemble_core_result, 6
bootstrap_variability, 7
bpparam_from_threads, 7
calibrateDeviations, 8
check_annotation_inputs, 9
check_core_inputs, 10
check_deviation_inputs, 10
check_enhancer, 11
check_file_run_inputs, 12
check_footprint_inputs, 12
check_gc_dist, 13
check_gcfreqs, 13
check_msites, 14
check_rnb_inputs, 14
check_run_options, 15
check_tf_bindsites, 15
check_variability_inputs, 16
computeExpectations, 18
computeExpectedFootprint, 18
computeFootprint, 19
computeRowZScore, 20
create_sink, 22
dev_helper, 25
diff_pvalues, 26
expected_footprint_plot, 27
footprint_difference, 27
granges_helper, 28
group_mean_difference, 28
kw_helper, 29
locate_sample_files, 29
methylTFR-package, 4
methyltfr_core, 30
normalise_footprint_flank, 31
parse_allc, 32
parse_bismarkcov, 32
parse_bismarkcytosine, 33
parse_bissnp, 33
parse_encode, 34
parse_epp, 34
process_core_sample, 38
read_sample_annotation, 41
resolve_annotation_file, 41
resolve_diff_groups, 42
resolve_rnb_sample_ann, 42
rnb_annotation_by_id, 43
rnb_annotation_table, 43
rnb_annotation_target, 44
rnb_column, 44
rnb_has_coverage, 45
rnb_has_dpval, 45
rnb_probe_ids, 46
rnb_quality_mode, 46
rnb_sample_ids, 47
rnb_sample_msites, 47
rnb_sites_to_granges, 48
set_grid, 53
t_helper, 53
valid_core_motifs, 54
variability_test, 54
wilcoxon_helper, 55
write_block_to_sink, 55

deviationZScores, *30*

addGCBinToMethylome, 5, *11*, *17*, *40*, *49*
anova_helper, 6
assemble_core_result, 6

bootstrap_variability, 7
bpparam_from_threads, 7

calibrateDeviations, 8
cbind, *39*
cbind,methylTFRdeviations-method, 8
check_annotation_inputs, 9
check_core_inputs, 10
check_deviation_inputs, 10
check_enhancer, 11
check_file_run_inputs, 12
check_footprint_inputs, 12
check_gc_dist, 13
check_gcfreqs, 13
check_msites, 14
check_rnb_inputs, 14
check_run_options, 15

check_tf_bindsites, 15
check_variability_inputs, 16
computeDeviation, *5, 6*, 16, *36*, *40*, *49*
computeExpectations, 18
computeExpectedFootprint, 18
computeFootprint, 19
computeRowZScore, 20
computeZScoreVariability, 20, *26*
computeZScoreVariability(), *4*
create_sink, 22

dev_helper, 25
deviations, 23, *30*, *50*
deviations(), *4*
deviations,methylTFRdeviations-method, 23
deviationZScores, *21*, 24, *50*
deviationZScores(), *4*
deviationZScores,methylTFRdeviations-method, 24
diff_pvalues, 26
differentialDeviationTest, *21*, 25
differentialDeviationTest(), *4*

expected_footprint_plot, 27

footprint_difference, 27

granges_helper, 28
group_mean_difference, 28

kw_helper, 29

locate_sample_files, 29

methylTFR *(*methylTFR-package*)*, 4
methylTFR-package, 4
methyltfr_core, 30
methylTFRdeviations, *25*, *49*, *52*
methylTFRdeviations-class, 30

normalise_footprint_flank, 31

p.adjust, *21*, *25*
parse_allc, 32
parse_bismarkcov, 32
parse_bismarkcytosine, 33
parse_bissnp, 33
parse_encode, 34
parse_epp, 34
plotExpectedFootprint, 35, *37*
plotExpectedFootprint(), *4*
plotMotifFootprint, *17*, *36*, 36
plotMotifFootprint(), *4*

process_core_sample, 38

rbind,methylTFRdeviations-method, 39
read_sample_annotation, 41
readMethylome, *5, 6*, *11*, *13*, *17*, *19*, *35*, *37*, 40, *49*
readMethylome(), *4*
resolve_annotation_file, 41
resolve_diff_groups, 42
resolve_rnb_sample_ann, 42
rnb_annotation_by_id, 43
rnb_annotation_table, 43
rnb_annotation_target, 44
rnb_column, 44
rnb_has_coverage, 45
rnb_has_dpval, 45
rnb_probe_ids, 46
rnb_quality_mode, 46
rnb_sample_ids, 47
rnb_sample_msites, 47
rnb_sites_to_granges, 48
runMethylTFR, *17*, *30*, *40*, 48, *52*
runMethylTFR(), *4*
runMethylTFRFromRnBeads, *30*, *50*, 50
runMethylTFRFromRnBeads(), *4*

set_grid, 53
SummarizedExperiment, *30*

t_helper, 53

valid_core_motifs, 54
variability_test, 54

wilcoxon_helper, 55
write_block_to_sink, 55