Package 'NanoMethViz'

October 13, 2026

Type Package

Title Visualise methylation data from Oxford Nanopore sequencing

Version 3.8.0

Description NanoMethViz is a toolkit for visualising methylation data from Oxford Nanopore sequencing. It can be used to explore methylation patterns from reads derived from Oxford Nanopore direct DNA sequencing with methylation called by callers including nanopolish, f5c and megalodon. The plots in this package allow the visualisation of methylation profiles aggregated over experimental groups and across classes of genomic features.

biocViews Software, LongRead, Visualization, DifferentialMethylation, DNAMethylation, Epigenetics, DataImport

URL https://github.com/shians/NanoMethViz, https://shians.github.io/NanoMethViz/

BugReports https://github.com/Shians/NanoMethViz/issues

Depends R (>= 4.0.0), methods, ggplot2 (>= 3.4.0)

Imports cpp11 (>= 0.2.5), readr, cli, S4Vectors, SummarizedExperiment, BiocSingular, bsseq, forcats, assertthat, AnnotationDbi, Rcpp, dplyr, dbscan, e1071, fs, GenomicRanges, Biostrings, ggrastr, glue, graphics, IRanges, limma (>= 3.44.0), patchwork, purrr, rlang, R.utils, Rsamtools, scales (>= 1.2.0), stats, stringr, tibble, tidyr, utils, withr

Suggests BiocStyle, Mus.musculus (>= 1.3.1), Homo.sapiens (>= 1.3.1), org.Hs.eg.db, TxDb.Hsapiens.UCSC.hg19.knownGene, TxDb.Hsapiens.UCSC.hg38.knownGene, org.Mm.eg.db, TxDb.Mmusculus.UCSC.mm10.knownGene, TxDb.Mmusculus.UCSC.mm39.refGene, knitr, rmarkdown, rtracklayer, testthat (>= 3.0.0), covr

LinkingTo Rcpp

License Apache License (>= 2.0)

SystemRequirements C++20

VignetteBuilder knitr

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/parallel true

Config/testthat/edition 3

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

git_branch RELEASE_3_23

git_last_commit 4193d2c

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-10-12

Author Shian Su [cre, aut]

Maintainer Shian Su <su.s@wehi.edu.au>

Contents

NanoMethViz-package . . . 3
assert_valid_exons . . . 4
assert_valid_genomic_coords . . . 4
assert_valid_methy_samples . . . 5
assert_valid_samples . . . 5
bsseq_to_edger . . . 6
bsseq_to_log_methy_ratio . . . 6
cluster_reads . . . 7
cluster_regions . . . 8
convert_methy_format . . . 8
create_tabix_file . . . 9
exons . . . 10
exons<- . . . 10
exons_to_genes . . . 10
filter_methy . . . 11
get_cgi . . . 12
get_cgi_mm10 . . . 12
get_example_exons_mus_musculus . . . 13
get_exons_homo_sapiens . . . 14
get_exons_mus_musculus . . . 14
load_example_modbamresult . . . 15
load_example_nanomethresult . . . 15
methy . . . 16
methy<- . . . 16
methy_col_names . . . 17
methy_to_bsseq . . . 17
methy_to_edger . . . 18
ModBamFiles . . . 18

ModBamFiles-class . . . 19
ModBamResult-class . . . 19
modbam_to_tabix . . . 21
mod_code . . . 22
mod_code<- . . . 22
NanoMethResult-class . . . 22
plot_agg_genes . . . 24
plot_agg_regions . . . 26
plot_gene . . . 27
plot_gene_heatmap . . . 29
plot_grange . . . 31
plot_grange_heatmap . . . 32
plot_mds . . . 33
plot_pca . . . 34
plot_region . . . 35
plot_region_heatmap . . . 39
plot_violin . . . 41
query_exons . . . 42
query_methy . . . 43
raw_methy_to_tabix . . . 44
reexports . . . 44
region_methy_stats . . . 45
samples . . . 45
samples<- . . . 46
sort_methy_file . . . 46

Index **47**

`NanoMethViz-package` *NanoMethViz: Visualise methylation data from Oxford Nanopore sequencing*

Description

NanoMethViz is a toolkit for visualising methylation data from Oxford Nanopore sequencing. It can be used to explore methylation patterns from reads derived from Oxford Nanopore direct DNA sequencing with methylation called by callers including nanopolish, f5c and megalodon. The plots in this package allow the visualisation of methylation profiles aggregated over experimental groups and across classes of genomic features.

Details

The main plotting functions in this package are `plot_gene()` and `plot_region()`.

- See `vignette("UserGuide", package = "NanoMethViz")` for documentation of how to use this package.

Author(s)

Maintainer: Shian Su <su.s@wehi.edu.au>

See Also

Useful links:

- https://github.com/shians/NanoMethViz
- https://shians.github.io/NanoMethViz/
- Report bugs at https://github.com/Shians/NanoMethViz/issues

---

`assert_valid_exons` *Validate exon annotation data frame*

Description

Validate exon annotation data frame

Usage

```
assert_valid_exons(exons)
```

Arguments

`exons`	exon annotation data.frame

`assert_valid_genomic_coords` *Assert that file paths are readable*

Description

This function checks whether all provided file paths exist and are readable. If any file paths do not exist, it throws an informative error message.

Usage

```
assert_valid_genomic_coords(chr, start, end, allow_equal = FALSE)
```

Arguments

`x`	A character vector of file paths to check for existence

Value

Nothing if all files exist, otherwise throws an error

assert_valid_methy_samples *Enhanced validation with better error messages*

Description

Enhanced validation with better error messages

Usage

```
assert_valid_methy_samples(data_path, samples, sample_check_limit = 1000)
```

Arguments

data_path	path to methylation data file
samples	sample annotation data.frame
sample_check_limit	number of entries to check for sample matching

assert_valid_samples *Validate sample annotation data frame*

Description

Validate sample annotation data frame

Usage

```
assert_valid_samples(
  samples,
  required_cols = c("sample", "group"),
  context = "sample annotation"
)
```

Arguments

samples	sample annotation data.frame
required_cols	required column names
context	description of where this is being used for error messages

bsseq_to_edger *Convert BSseq object to edgeR methylation matrix*

Description

Convert BSseq object to edgeR methylation matrix

Usage

```
bsseq_to_edger(bsseq, regions = NULL)
```

Arguments

bsseq	the BSseq object.
regions	the regions to calculate log-methylation ratios over. If left NULL, ratios will be calculated per site.

Value

a matrix compatible with the edgeR differential methylation pipeline

Examples

```
methy <- system.file("methy_subset.tsv.bgz", package = "NanoMethViz", mustWork = FALSE)
bsseq <- methy_to_bsseq(methy)
edger_mat <- bsseq_to_edger(bsseq)
```

bsseq_to_log_methy_ratio *Convert BSseq object to log-methylation-ratio matrix*

Description

Creates a log-methylation-ratio matrix from a BSseq object that is useful for dimensionality reduction plots.

Usage

```
bsseq_to_log_methy_ratio(
  bsseq,
  regions = NULL,
  prior_count = 2,
  drop_na = TRUE
)
```

Arguments

bsseq	the BSseq object.
regions	the regions to calculate log-methylation ratios over. If left NULL, ratios will be calculated per site.
prior_count	the prior count added to avoid taking log of 0.
drop_na	whether to drop rows with all NA values.

Value

a matrix containing log-methylation-ratios.

Examples

```
nmr <- load_example_nanomethresult()
bsseq <- methy_to_bsseq(nmr)
regions <- exons_to_genes(NanoMethViz::exons(nmr))
log_m_ratio <- bsseq_to_log_methy_ratio(bsseq, regions)
```

---

cluster_reads *Cluster reads based on methylation*

Description

Cluster reads based on methylation

Usage

```
cluster_reads(x, chr, start, end, min_pts = 5)
```

Arguments

x	a ModBamResult object.
chr	the chromosome name where to find the region.
start	the start position of the region.
end	the end position of the region.
min_pts	the minimum number of points needed to form a cluster (default = 10).

Value

A tibble with information about each read's cluster assignment and read statistics.

cluster_regions *Cluster regions by K-means*

Description

Cluster regions by k-means based on their methylation profiles. In order to cluster using k-means the methylation profile of each region is interpolated and sampled at fixed points. The first 10 principal components are used for the k-means clustering. The clustering is best behaved in regions of similar width and CpG density.

Usage

```
cluster_regions(x, regions, centers = 2, grid_method = c("density", "uniform"))
```

Arguments

x	the NanoMethResult object.
regions	a table of regions containing at least columns chr, strand, start and end.
centers	number of centers for k-means, identical to the number of output clusters.
grid_method	the method for generating the sampling grid. The default option "density" attempts to create a grid with similar density as the data, "uniform" creates a grid of uniform density.

Value

the table of regions given by the 'regions' argument with the column 'cluster' added.

Examples

```
nmr <- load_example_nanomethresult()
gene_anno <- exons_to_genes(NanoMethViz::exons(nmr))
# uniform grid due to low number of input features
gene_anno_clustered <- cluster_regions(nmr, gene_anno, centers = 2, grid_method = "uniform")
plot_agg_regions(nmr, gene_anno_clustered, group_col = "cluster")
```

convert_methy_format *Convert methylation calls to NanoMethViz format*

Description

Convert methylation calls to NanoMethViz format

Usage

```
convert_methy_format(
  input_files,
  output_file,
  samples = fs::path_ext_remove(fs::path_file(input_files)),
  verbose = TRUE
)
```

Arguments

input_files	the files to convert
output_file	the output file to write results to (must end in .bgz)
samples	the names of samples corresponding to each file
verbose	TRUE if progress messages are to be printed

Value

invisibly returns the output file path, creates a tabix file (.bgz) and its index (.bgz.tbi)

---

create_tabix_file *Create a tabix file using methylation calls*

Description

Create a tabix file using methylation calls

Usage

```
create_tabix_file(
  input_files,
  output_file,
  samples = extract_file_names(input_files),
  verbose = TRUE
)
```

Arguments

input_files	the files to convert
output_file	the output file to write results to (must end in .bgz)
samples	the names of samples corresponding to each file
verbose	TRUE if progress messages are to be printed

Value

invisibly returns the output file path, creates a tabix file (.bgz) and its index (.bgz.tbi)

Examples

```
methy_calls <- system.file(package = "NanoMethViz",
    c("sample1_nanopolish.tsv.gz", "sample2_nanopolish.tsv.gz"), mustWork = FALSE)
temp_file <- paste0(tempfile(), ".tsv.bgz")

create_tabix_file(methy_calls, temp_file)
```

---

exons *Get exon annotation*

Description

Get exon annotation

Usage

```
exons(object)
```

---

exons<- *Set exon annotation*

Description

Set exon annotation

Usage

```
exons(object) <- value
```

---

exons_to_genes *Convert exon annotation to genes*

Description

Convert exon annotation to genes

Usage

```
exons_to_genes(x)
```

Arguments

x	the exon level annotation containing columns "gene_id", "chr", "strand" and "symbol".

Value

the gene level annotation where each gene is taken to span the earliest start position and latest end position of its exons.

Examples

```
nmr <- load_example_nanomethresult()
exons_to_genes(NanoMethViz::exons(nmr))
```

---

filter_methy *Create filtered methylation file*

Description

Create a filtered methylation file from an existing one.

Usage

```
filter_methy(x, output_file, ...)
```

Arguments

x	the path to the methylation file or a NanoMethResult object.
output_file	the output file to write results to (must end in .bgz).
...	filtering criteria given in dplyr syntax. Use methy_col_names() to get available column names.

Value

invisibly returns 'output_file' if x is a file path, otherwise returns NanoMethResult object with methy(x) replaced with filtered value.

Examples

```
nmr <- load_example_nanomethresult()
output_file <- paste0(tempfile(), ".tsv.bgz")
filter_methy(nmr, output_file = output_file, chr == "chrX")
filter_methy(methy(nmr), output_file = output_file, chr == "chrX")
```

get_cgi *Get CpG islands annotations*

Description

Helper functions are provided for obtaining CpG islands annotations from UCSC table browser.

Usage

```
get_cgi(genome)
```

Value

tibble (data.frame) object containing CpG islands annotation.

get_cgi_mm10 *Get exon annotations*

Description

Helper functions are provided for obtaining exon annotations from relevant TxDb packages on Bioconductor for the construction of NanoMethResult objects.

Usage

```
get_cgi_mm10()

get_cgi_grcm39()

get_cgi_t2t()

get_cgi_hg19()

get_cgi_hg38()

get_exons_mm10()

get_exons_grcm39()

get_exons_hg19()

get_exons_hg38()

get_exons_t2t()
```

Value

tibble (data.frame) object containing exon annotation.

Examples

```
cgi_mm10 <- get_cgi_mm10()

cgi_GRCm39 <- get_cgi_grcm39()

cgi_t2t <- get_cgi_t2t()

cgi_hg19 <- get_cgi_hg19()

cgi_hg38 <- get_cgi_hg38()

mm10_exons <- get_exons_mm10()

grcm39_exons <- get_exons_grcm39()

hg19_exons <- get_exons_hg19()

hg38_exons <- get_exons_hg38()

t2t_exons <- get_exons_t2t()
```

---

get_example_exons_mus_musculus

Get example exon annotations for mus musculus (mm10)

Description

This is a small subset of the exons returned by `get_exons_mus_musculus()` for demonstrative purposes. It contains the exons for the genes Brca1, Brca2, Impact, Meg3, Peg3 and Xist.

Usage

```
get_example_exons_mus_musculus()
```

Value

data.frame containing exons

Examples

```
example_exons <- get_example_exons_mus_musculus()
```

get_exons_homo_sapiens

Get exon annotations for Homo sapiens (hg19)

Description

Get exon annotations for Homo sapiens (hg19)

Usage

```
get_exons_homo_sapiens()
```

Value

data.frame containing exons

Examples

```
h_sapiens_exons <- get_exons_homo_sapiens()
```

get_exons_mus_musculus

Get exon annotations for Mus musculus (mm10)

Description

Get exon annotations for Mus musculus (mm10)

Usage

```
get_exons_mus_musculus()
```

Value

data.frame containing exons

Examples

```
m_musculus_exons <- get_exons_mus_musculus()
```

load_example_modbamresult *Load an example ModBamResult object*

Description

Load an example ModBamResult object for demonstration of plotting functions. Run `load_example_modbamresult` without the function call to see how the object is constructed.

Usage

```
load_example_modbamresult()
```

Value

a ModBamResult object

Examples

```
mbr <- load_example_modbamresult()
```

load_example_nanomethresult *Load an example NanoMethResult object*

Description

Load an example NanoMethResult object for demonstration of plotting functions. Run `load_example_nanomethresult` without the function call to see how the object is constructed.

Usage

```
load_example_nanomethresult()
```

Value

a NanoMethResult object

Examples

```
nmr <- load_example_nanomethresult()
```

methy *Get methylation data*

Description

Get methylation data

Usage

```
methy(object)
```

Arguments

object	the object.

Value

the path to the methylation data.

Examples

```
showMethods("methy")
```

methy<- *Set methylation data*

Description

Set methylation data

Usage

```
methy(object) <- value
```

methy_col_names *Column names for methylation data*

Description

Column names for methylation data

Usage

```
methy_col_names()
```

Value

column names for methylation data

Examples

```
methy_col_names()
```

methy_to_bsseq *Create BSSeq object from methylation tabix file*

Description

Create BSSeq object from methylation tabix file

Usage

```
methy_to_bsseq(methy, out_folder = tempdir(), verbose = TRUE)
```

Arguments

methy	the NanoMethResult object or path to the methylation tabix file.
out_folder	the folder to store intermediate files. One file is created for each sample and contains columns "chr", "pos", "total" and "methylated".
verbose	TRUE if progress messages are to be printed

Value

a BSSeq object.

Examples

```
nmr <- load_example_nanomethresult()
bsseq <- methy_to_bsseq(nmr)
```

methy_to_edger *Convert NanoMethResult object to edgeR methylation matrix*

Description

Convert NanoMethResult object to edgeR methylation matrix

Usage

```
methy_to_edger(methy, regions = NULL, out_folder = tempdir(), verbose = TRUE)
```

Arguments

methy	the NanoMethResult object or path to the methylation tabix file.
regions	the regions to calculate log-methylation ratios over. If left NULL, ratios will be calculated per site.
out_folder	the folder to store intermediate files. One file is created for each sample and contains columns "chr", "pos", "total" and "methylated".
verbose	TRUE if progress messages are to be printed

Value

a matrix compatible with the edgeR differential methylation pipeline

Examples

```
nmr <- load_example_nanomethresult()
edger_mat <- methy_to_edger(nmr)
```

ModBamFiles *Constructor for a ModBamFiles object*

Description

This function creates a ModBamFiles object containing information about the samples and file paths. This constructor checks that the files are readable and have an index.

Usage

```
ModBamFiles(samples, paths)

## S4 method for signature 'ModBamFiles'
show(object)
```

Arguments

samples	a character vector with the names of the samples.
paths	a character vector with the file paths for the BAM files.
object	a ModBamFiles object.

Value

A ModBamFiles object with the sample and path information.

ModBamFiles-class *ModBamFiles class*

Description

This is a class for holding information about modBAM files. It is a data.frame containing information about samples and paths to modBAM files.

ModBamResult-class *modBAM methylation results*

Description

A ModBamResult object stores modBAM data used for NanoMethViz visualisation. It contains stores a ModBamFiles object, sample information and optional exon information. The object is constructed using the ModBamResult() constructor function described in "Usage".

Usage

```
## S4 method for signature 'ModBamResult'
methy(object)

## S4 replacement method for signature 'ModBamResult,ModBamFiles'
methy(object) <- value

## S4 method for signature 'ModBamResult'
samples(object)

## S4 replacement method for signature 'ModBamResult,data.frame'
samples(object) <- value

## S4 method for signature 'ModBamResult'
exons(object)

## S4 replacement method for signature 'ModBamResult,data.frame'
```

```
exons(object) <- value

## S4 method for signature 'ModBamResult'
mod_code(object)

## S4 replacement method for signature 'ModBamResult,character'
mod_code(object) <- value

ModBamResult(methy, samples, exons = NULL, mod_code = "m")
```

Arguments

object	the ModBamResult object.
value	the mod code.
methy	a ModBamFiles object.
samples	the data.frame of sample annotation containing at least columns sample and group.
exons	(optional) the data.frame of exon information containing at least columns gene_id, chr, strand, start, end, transcript_id and symbol.
mod_code	a character with the mod code of interest. Defaults to "m" for 5mC. See details for other options.

Details

The possible tags for mod_code can be found at https://samtools.github.io/hts-specs/SAMtags.pdf under the 'Base modifications' section.

Value

a ModBamResult object to be used with plotting functions

a ModBamFiles data.frame.

the sample annotation.

the exon annotation.

the mod code.

Functions

- `methy(ModBamResult)`: modBAM information getter.
- `methy(object = ModBamResult) <- value`: modBAM information setter.
- `samples(ModBamResult)`: sample annotation getter.
- `samples(object = ModBamResult) <- value`: sample annotation setter.
- `exons(ModBamResult)`: exon annotation getter.
- `exons(object = ModBamResult) <- value`: exon annotation setter.
- `mod_code(ModBamResult)`: mod code getter.
- `mod_code(object = ModBamResult) <- value`: mod code setter.
- `ModBamResult()`: Constructor

Slots

`methy` a ModBamFiles data.frame specifying the samples and paths to bam files.

`samples` the data.frame of sample annotation containing at least columns sample and group.

`exons` the data.frame of exon information containing at least columns gene_id, chr, strand, start, end, transcript_id and symbol.

`mod_code` the modification code of interest.

`modbam_to_tabix` *Convert BAM with modifications to tabix format*

Description

The `modbam_to_tabix` function takes a ModBamResult object and converts it into a tabix file format, which is efficient for indexing and querying large datasets.

Usage

```
modbam_to_tabix(x, out_file, mod_code = NanoMethViz::mod_code(x))
```

Arguments

`x`	the `ModBamResult` object.
`out_file`	the path of the output tabix.
`mod_code`	the modification code to use, defaults to 'm' for 5mC methylation.

Details

The possible tags for mod_code can be found at https://samtools.github.io/hts-specs/SAMtags.pdf under the 'Base modifications' section.

Value

invisibly returns the name of the created tabix file.

Examples

```
out_file <- paste0(tempfile(), ".tsv.bgz")
mbr <- ModBamResult(
    methy = ModBamFiles(
        samples = "sample1",
        paths = system.file("peg3.bam", package = "NanoMethViz",
        mustWork = FALSE)
    ),
    samples = data.frame(
        sample = "sample1",
        group = "group1"
    )
```

```
)

modbam_to_tabix(mbr, out_file)
```

`mod_code` *Get mod code*

Description

Get mod code

Usage

```
mod_code(object)
```

`mod_code<-` *Set mod code*

Description

Set mod code

Usage

```
mod_code(object) <- value
```

`NanoMethResult-class` *Nanopore Methylation Result*

Description

A NanoMethResult object stores data used for NanoMethViz visualisation. It contains stores a path to the methylation data, sample information and optional exon information. The object is constructed using the NanoMethResult() constructor function described in "Usage".

Usage

```
NanoMethResult(methy, samples, exons = NULL)

## S4 method for signature 'NanoMethResult'
methy(object)

## S4 replacement method for signature 'NanoMethResult,ANY'
methy(object) <- value

## S4 method for signature 'NanoMethResult'
samples(object)

## S4 replacement method for signature 'NanoMethResult,data.frame'
samples(object) <- value

## S4 method for signature 'NanoMethResult'
exons(object)

## S4 replacement method for signature 'NanoMethResult,data.frame'
exons(object) <- value
```

Arguments

methy	the path to the methylation tabix file.
samples	the data.frame of sample annotation containing at least columns sample and group.
exons	(optional) the data.frame of exon information containing at least columns gene_id, chr, strand, start, end, transcript_id and symbol.
object	the NanoMethResult object.
value	the exon annotation.

Value

a NanoMethResult object to be used with plotting functions

the path to the methylation data.

the sample annotation.

the exon annotation.

Functions

- `NanoMethResult()`: Constructor
- `methy(NanoMethResult)`: methylation data path getter.
- `methy(object = NanoMethResult) <- value`: methylation data path setter.
- `samples(NanoMethResult)`: sample annotation getter.
- `samples(object = NanoMethResult) <- value`: sample annotation setter.

- `exons(NanoMethResult)`: exon annotation getter.
- `exons(object = NanoMethResult) <- value`: exon annotation setter.

Slots

`methy` the path to the methylation tabix file.

`samples` the data.frame of sample annotation containing at least columns sample and group.

`exons` the data.frame of exon information containing at least columns gene_id, chr, strand, start, end, transcript_id and symbol.

Examples

```
methy <- system.file(package = "NanoMethViz", "methy_subset.tsv.bgz", mustWork = FALSE)
sample <- c(
    "B6Cast_Prom_1_bl6",
    "B6Cast_Prom_1_cast",
    "B6Cast_Prom_2_bl6",
    "B6Cast_Prom_2_cast",
    "B6Cast_Prom_3_bl6",
    "B6Cast_Prom_3_cast"
)
group <- c(
    "bl6",
    "cast",
    "bl6",
    "cast",
    "bl6",
    "cast"
)
sample_anno <- data.frame(sample, group, stringsAsFactors = FALSE)
exon_tibble <- get_example_exons_mus_musculus()
NanoMethResult(methy, sample_anno, exon_tibble)

x <- load_example_nanomethresult()
methy(x)
```

---

plot_agg_genes *Plot gene aggregate plot*

Description

Plot gene aggregate plot

Usage

```
plot_agg_genes(
  x,
  genes = NULL,
  binary_threshold = 0.5,
  group_col = NULL,
  flank = 2000,
  stranded = TRUE,
  span = 0.05,
  palette = ggplot2::scale_colour_brewer(palette = "Set1")
)
```

Arguments

x	the NanoMethResult or ModBamResult object.
genes	a character vector of gene symbols to include in aggregate plot. If NULL (default), all genes in exons(x) are used.
binary_threshold	the modification probability such that calls with modification probability above the threshold are considered methylated, and those with probability equal or below are considered unmethylated.
group_col	the column name to group aggregated trends by. This column can be found in either the regions table or samples(x). When NULL (default), all data is aggregated together. Common values include "sample" to show individual samples or "group" to show sample groups.
flank	the number of flanking bases to add to each side of each region.
stranded	if TRUE, negative strand features will have their coordinates flipped to reflect biological features like transcription start sites (e.g., for genes, coordinates run from TSS to TES regardless of strand).
span	the span parameter for loess smoothing of the trend lines.
palette	the ggplot colour palette used for groups.

Details

This function creates an aggregate methylation profile across multiple genes by scaling all genes to the same relative coordinates (0 to 1) and averaging methylation levels at each relative position. Genes are optionally extended by flanking regions specified by the flank parameter. The resulting plot shows smoothed trends of average methylation probability from gene start to gene end, with optional flanking regions.

Value

a ggplot object containing the aggregate methylation trend of genes.

Examples

```
nmr <- load_example_nanomethresult()
plot_agg_genes(nmr)

# Plot specific genes only
plot_agg_genes(nmr, genes = c("Peg3", "Impact"))

# Group by sample
plot_agg_genes(nmr, group_col = "sample")
```

`plot_agg_regions` *Plot aggregate regions*

Description

Plot aggregate regions

Usage

```
plot_agg_regions(
  x,
  regions,
  binary_threshold = 0.5,
  group_col = NULL,
  flank = 2000,
  stranded = TRUE,
  span = 0.05,
  palette = ggplot2::scale_colour_brewer(palette = "Set1")
)
```

Arguments

`x`	the NanoMethResult or ModBamResult object.
`regions`	a table of regions containing at least columns chr, strand, start and end. Any additional columns can be used for grouping.
`binary_threshold`	the modification probability such that calls with modification probability above the threshold are considered methylated, and those with probability equal or below are considered unmethylated.
`group_col`	the column name to group aggregated trends by. This column can be found in either the regions table or samples(x). When NULL (default), all data is aggregated together. Common values include "sample" to show individual samples or "group" to show sample groups.
`flank`	the number of flanking bases to add to each side of each region.

stranded	if TRUE, negative strand features will have their coordinates flipped to reflect biological features like transcription start sites (e.g., for genes, coordinates run from TSS to TES regardless of strand).
span	the span parameter for loess smoothing of the trend lines.
palette	the ggplot colour palette used for groups.

Value

a ggplot object containing the aggregate methylation trend.

Examples

```
nmr <- load_example_nanomethresult()
gene_anno <- exons_to_genes(NanoMethViz::exons(nmr))
plot_agg_regions(nmr, gene_anno)
plot_agg_regions(nmr, gene_anno, group_col = "sample")
plot_agg_regions(nmr, gene_anno, group_col = "group")
```

---

plot_gene *Plot gene methylation*

Description

Plot the methylation of a gene symbol specified within the exon(x) slot.

Usage

```
plot_gene(x, gene, ...)

## S4 method for signature 'NanoMethResult,character'
plot_gene(
  x,
  gene,
  window_prop = 0.3,
  anno_regions = NULL,
  binary_threshold = NULL,
  avg_method = c("mean", "median"),
  spaghetti = FALSE,
  heatmap = TRUE,
  heatmap_subsample = 50,
  smoothing_window = 2000,
  gene_anno = TRUE,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  mod_scale = c(0, 1),
  span = NULL
```

```
)

## S4 method for signature 'ModBamResult,character'
plot_gene(
  x,
  gene,
  window_prop = 0.3,
  anno_regions = NULL,
  binary_threshold = NULL,
  avg_method = c("mean", "median"),
  spaghetti = FALSE,
  heatmap = TRUE,
  heatmap_subsample = 50,
  smoothing_window = 2000,
  gene_anno = TRUE,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  mod_scale = c(0, 1),
  span = NULL
)
```

Arguments

x	the NanoMethResult or ModBamResult object.
gene	the gene symbol for the gene to plot.
...	additional arguments.
window_prop	the size of flanking region to plot. Can be a vector of two values for left and right window size. Values indicate proportion of gene length.
anno_regions	the data.frame of regions to be annotated.
binary_threshold	the modification probability such that calls with modification probability above the threshold are set to 1 and probabilities equal to or below the threshold are set to 0.
avg_method	the average method for pre-smoothing at each genomic position. Data is pre-smoothed at each genomic position before the smoothed aggregate line is generated for performance reasons. The default is "mean" which corresponds to the average methylation fraction. The alternative "median" option is closer to an average within the more common methylation state.
spaghetti	whether or not individual reads should be shown.
heatmap	whether or not read-methylation heatmap should be shown.
heatmap_subsample	how many packed rows of reads to subsample to.
smoothing_window	the window size for smoothing the trend line.
gene_anno	whether to show gene annotation.

palette	the ggplot colour palette used for groups.
line_size	the size of the lines.
mod_scale	the scale range for modification probabilities. Default c(0, 1), set to "auto" for automatic limits.
span	DEPRECATED, use smoothing_window instead. Will be removed in next version.

Details

This function plots the methylation data for a given gene. The main trendline plot shows the average methylation probability across the gene. The heatmap plot shows the methylation probability for each read across the gene. The gene annotation plot shows the exons of the gene. In the heatmap, each row represents one or more non-overlapping reads where the coloured segments represent the methylation probability at each position. Data along a read is connected by a grey line. The gene annotation plot shows the isoforms and exons of the gene, with arrows indicating the direction of transcription.

Since V3.0.0 NanoMethViz has changed the smoothing strategy from a loess smoothing to a weighted moving average. This is because the loess smoothing was too computationally expensive for large datasets and had a span parameter that was difficult to tune. The new smoothing strategy is controlled by the smoothing_window argument.

Value

a patchwork plot containing the methylation profile in the specified region.

Functions

- `plot_gene(x = ModBamResult, gene = character)`: S4 method for ModBamResult

Examples

```
nmr <- load_example_nanomethresult()
plot_gene(nmr, "Peg3")
```

---

plot_gene_heatmap *Plot gene methylation heatmap*

Description

Plot the methylation heatmap of a gene symbol specified within the exon(x) slot.

Usage

```
plot_gene_heatmap(x, gene, ...)

## S4 method for signature 'NanoMethResult,character'
plot_gene_heatmap(
  x,
  gene,
  window_prop = 0.3,
  pos_style = c("to_scale", "compact"),
  subsample = 50
)

## S4 method for signature 'ModBamResult,character'
plot_gene_heatmap(
  x,
  gene,
  window_prop = 0.3,
  pos_style = c("to_scale", "compact"),
  subsample = 50
)
```

Arguments

x	the NanoMethResult or ModBamResult object.
gene	the gene symbol for the gene to plot.
...	additional arguments.
window_prop	the size of flanking region to plot. Can be a vector of two values for left and right window size. Values indicate proportion of gene length.
pos_style	the style for plotting the base positions along the x-axis. Defaults to "to_scale", plotting (potentially) overlapping squares along the genomic position to scale. The "compact" options plots only the positions with measured modification.
subsample	the number of read of packed read rows to subsample to.

Details

This function creates a heatmap visualisation of methylation data for a specific gene. Each row in the heatmap represents one or more packed reads, where colored segments indicate methylation probability at each genomic position.

Value

a ggplot object of the heatmap

a ggplot plot containing the heatmap.

Examples

```
nmr <- load_example_nanomethresult()
plot_gene_heatmap(nmr, "Peg3")
```

`plot_grange` *Plot GRanges*

Description

Plot GRanges

Usage

```
plot_grange(
  x,
  grange,
  anno_regions = NULL,
  binary_threshold = NULL,
  avg_method = c("mean", "median"),
  spaghetti = FALSE,
  heatmap = TRUE,
  heatmap_subsample = 50,
  gene_anno = TRUE,
  smoothing_window = 2000,
  window_prop = 0,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  span = NULL
)
```

Arguments

`x`	the NanoMethResult object.
`grange`	the GRanges object with one entry.
`anno_regions`	the data.frame of regions to be annotated.
`binary_threshold`	the modification probability such that calls with modification probability above the threshold are set to 1 and probabilities equal to or below the threshold are set to 0.
`avg_method`	the average method for pre-smoothing at each genomic position. Data is pre-smoothed at each genomic position before the smoothed aggregate line is generated for performance reasons. The default is "mean" which corresponds to the average methylation fraction. The alternative "median" option is closer to an average within the more common methylation state.
`spaghetti`	whether or not individual reads should be shown.

heatmap	whether or not read-methylation heatmap should be shown.
heatmap_subsample	how many packed rows of reads to subsample to.
gene_anno	whether to show gene annotation.
smoothing_window	the window size for smoothing the trend line.
window_prop	the size of flanking region to plot. Can be a vector of two values for left and right window size. Values indicate proportion of gene length.
palette	the ggplot colour palette used for groups.
line_size	the size of the lines.
span	DEPRECATED, use smoothing_window instead. Will be removed in next version.

Value

a patchwork plot containing the methylation profile in the specified region.

Examples

```
nmr <- load_example_nanomethresult()
plot_grange(nmr, GenomicRanges::GRanges("chr7:6703892-6730431"))
```

---

plot_grange_heatmap *Plot GRanges heatmap*

Description

Plot GRanges heatmap

Usage

```
plot_grange_heatmap(
  x,
  grange,
  pos_style = c("to_scale", "compact"),
  window_prop = 0,
  subsample = 50
)
```

Arguments

x	the NanoMethResult object.
grange	the GRanges object with one entry.
pos_style	the style for plotting the base positions along the x-axis. Defaults to "to_scale", plotting (potentially) overlapping squares along the genomic position to scale. The "compact" options plots only the positions with measured modification.
window_prop	the size of flanking region to plot. Can be a vector of two values for left and right window size. Values indicate proportion of region length.
subsample	the number of read of packed read rows to subsample to.

Value

a ggplot plot containing the heatmap.

Examples

```
nmr <- load_example_nanomethresult()
plot_grange_heatmap(nmr, GenomicRanges::GRanges("chr7:6703892-6730431"))
```

---

plot_mds *Plot MDS*

Description

Plot multi-dimensional scaling plot using algorithm of limma::plotMDS(). It is recommended this be done with the log-methylation-ratio matrix generated by bsseq_to_log_methy_ratio().

Usage

```
plot_mds(
  x,
  top = 500,
  plot_dims = c(1, 2),
  labels = colnames(x),
  groups = NULL,
  legend_name = "group"
)
```

Arguments

x	a numeric matrix of log-methylation-ratio values where rows are genomic regions/features and columns are samples.
top	the number of top variable features (genes/regions) used to calculate pairwise distances between samples.

plot_dims	the numeric vector of the two dimensions to be plotted (default c(1, 2) for first two principal coordinates).
labels	the character vector of labels for data points. By default uses column names of x, set to NULL to plot unlabeled points.
groups	the character vector or numeric vector of groups that data points will be coloured by. For character vectors, discrete colours are used. For numeric vectors, continuous colour scale is used. By default, groups is NULL and points are not coloured.
legend_name	the name for the colour legend.

Details

Multi-dimensional scaling (MDS) is a dimensionality reduction technique that represents high-dimensional distances between samples in a lower-dimensional space (typically 2D). This function uses limma's plotMDS algorithm, which selects the most variable features and calculates leading log-fold-change distances between samples. The resulting plot shows samples as points where proximity indicates similarity in methylation profiles. When variance explained is available, it is shown as percentages on the axis labels.

Value

ggplot object of the MDS plot.

Examples

```
nmr <- load_example_nanomethresult()
bss <- methy_to_bsseq(nmr)
lmr <- bsseq_to_log_methy_ratio(bss)
plot_mds(lmr)
```

---

plot_pca *Plot PCA*

Description

Plot principal component analysis plot using BiocSingular::runPCA(). It is recommended this be done with the log-methylation-ratio matrix generated by bsseq_to_log_methy_ratio().

Usage

```
plot_pca(
  x,
  plot_dims = c(1, 2),
  labels = colnames(x),
  groups = NULL,
  legend_name = "group"
)
```

Arguments

x	a numeric matrix where rows are genomic regions/features and columns are samples. Should be a log-methylation-ratio matrix for best results.
plot_dims	the numeric vector of the two dimensions to be plotted (default c(1, 2) for PC1 vs PC2).
labels	the character vector of labels for data points. By default uses column names of x, set to NULL to plot unlabeled points.
groups	the character vector or numeric vector of groups that data points will be coloured by. For character vectors, discrete colours are used. For numeric vectors, continuous colour scale is used. By default, groups is NULL and points are not coloured.
legend_name	the name for the colour legend.

Details

Principal Component Analysis (PCA) is a dimensionality reduction technique that identifies the directions of maximum variance in high-dimensional data. Unlike MDS, PCA axis labels show the principal component numbers rather than variance explained percentages. The first principal component (PC1) captures the most variance in the data, PC2 captures the second most, and so on. Samples that cluster together have similar methylation profiles across the genomic regions included in the analysis.

Value

ggplot object of the PCA plot.

Examples

```
nmr <- load_example_nanomethresult()
bss <- methy_to_bsseq(nmr)
lmr <- bsseq_to_log_methy_ratio(bss)
plot_pca(lmr)
```

---

plot_region *Plot region methylation*

Description

Plot the methylation of a genomic region.

Usage

```
plot_region(x, chr, start, end, ...)

## S4 method for signature 'NanoMethResult,character,numeric,numeric'
plot_region(
  x,
  chr,
  start,
  end,
  anno_regions = NULL,
  binary_threshold = NULL,
  avg_method = c("mean", "median"),
  spaghetti = FALSE,
  heatmap = TRUE,
  heatmap_subsample = 50,
  smoothing_window = 2000,
  gene_anno = TRUE,
  window_prop = 0,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  mod_scale = c(0, 1),
  span = NULL
)

## S4 method for signature 'ModBamResult,character,numeric,numeric'
plot_region(
  x,
  chr,
  start,
  end,
  anno_regions = NULL,
  binary_threshold = NULL,
  avg_method = c("mean", "median"),
  spaghetti = FALSE,
  heatmap = TRUE,
  heatmap_subsample = 50,
  smoothing_window = 2000,
  gene_anno = TRUE,
  window_prop = 0,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  mod_scale = c(0, 1),
  span = NULL
)

## S4 method for signature 'NanoMethResult,factor,numeric,numeric'
plot_region(
  x,
```

```
  chr,
  start,
  end,
  anno_regions = NULL,
  binary_threshold = NULL,
  avg_method = c("mean", "median"),
  spaghetti = FALSE,
  heatmap = TRUE,
  heatmap_subsample = 50,
  smoothing_window = 2000,
  gene_anno = TRUE,
  window_prop = 0,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  mod_scale = c(0, 1),
  span = NULL
)

## S4 method for signature 'ModBamResult,factor,numeric,numeric'
plot_region(
  x,
  chr,
  start,
  end,
  anno_regions = NULL,
  binary_threshold = NULL,
  avg_method = c("mean", "median"),
  spaghetti = FALSE,
  heatmap = TRUE,
  heatmap_subsample = 50,
  smoothing_window = 2000,
  gene_anno = TRUE,
  window_prop = 0,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  mod_scale = c(0, 1),
  span = NULL
)
```

Arguments

x	the NanoMethResult or ModBamResult object.
chr	the chromosome to plot.
start	the start of the plotting region.
end	the end of the plotting region.
...	additional arguments.
anno_regions	the data.frame of regions to be annotated.

`binary_threshold`
: the modification probability such that calls with modification probability above the threshold are set to 1 and probabilities equal to or below the threshold are set to 0.

`avg_method`
: the average method for pre-smoothing at each genomic position. Data is pre-smoothed at each genomic position before the smoothed aggregate line is generated for performance reasons. The default is "mean" which corresponds to the average methylation fraction. The alternative "median" option is closer to an average within the more common methylation state.

`spaghetti`
: whether or not individual reads should be shown.

`heatmap`
: whether or not read-methylation heatmap should be shown.

`heatmap_subsample`
: how many packed rows of reads to subsample to.

`smoothing_window`
: the window size for smoothing the trend line.

`gene_anno`
: whether to show gene annotation.

`window_prop`
: the size of flanking region to plot. Can be a vector of two values for left and right window size. Values indicate proportion of gene length.

`palette`
: the ggplot colour palette used for groups.

`line_size`
: the size of the lines.

`mod_scale`
: the scale range for modification probabilities. Default c(0, 1), set to "auto" for automatic limits.

`span`
: DEPRECATED, use smoothing_window instead. Will be removed in next version.

Details

This function plots the methylation data for a given region. The main trendline plot shows the average methylation probability across the region. The heatmap plot shows the methylation probability for each read across the region. The gene annotation plot shows the exons of the region. In the heatmap, each row represents one or more non-overlapping reads where the coloured segments represent the methylation probability at each position. Data along a read is connected by a grey line. The gene annotation plot shows the isoforms and exons of genes within the region, with arrows indicating the direction of transcription.

Since V3.0.0 NanoMethViz has changed the smoothing strategy from a loess smoothing to a weighted moving average. This is because the loess smoothing was too computationally expensive for large datasets and had a span parameter that was difficult to tune. The new smoothing strategy is controlled by the smoothing_window argument.

Value

a patchwork plot containing the methylation profile in the specified region.

Examples

```
nmr <- load_example_nanomethresult()
plot_region(nmr, "chr7", 6703892, 6730431)
```

---

plot_region_heatmap *Plot region methylation heatmap*

Description

Plot the methylation heatmap of a genomic region.

Usage

```
plot_region_heatmap(x, chr, start, end, ...)

## S4 method for signature 'NanoMethResult,character,numeric,numeric'
plot_region_heatmap(
  x,
  chr,
  start,
  end,
  pos_style = c("to_scale", "compact"),
  window_prop = 0,
  subsample = 50
)

## S4 method for signature 'ModBamResult,character,numeric,numeric'
plot_region_heatmap(
  x,
  chr,
  start,
  end,
  pos_style = c("to_scale", "compact"),
  window_prop = 0,
  subsample = 50
)

## S4 method for signature 'NanoMethResult,factor,numeric,numeric'
plot_region_heatmap(
  x,
  chr,
  start,
  end,
  pos_style = c("to_scale", "compact"),
  window_prop = 0,
  subsample = 50
```

```
)

## S4 method for signature 'ModBamResult,factor,numeric,numeric'
plot_region_heatmap(
  x,
  chr,
  start,
  end,
  pos_style = c("to_scale", "compact"),
  window_prop = 0,
  subsample = 50
)
```

Arguments

x	the NanoMethResult or ModBamResult object.
chr	the chromosome to plot.
start	the start of the plotting region.
end	the end of the plotting region.
...	additional arguments.
pos_style	the style for plotting the base positions along the x-axis. Defaults to "to_scale", plotting (potentially) overlapping squares along the genomic position to scale. The "compact" options plots only the positions with measured modification.
window_prop	the size of flanking region to plot. Can be a vector of two values for left and right window size. Values indicate proportion of region length.
subsample	the number of read of packed read rows to subsample to.

Details

This function creates a heatmap visualisation of methylation data for a specified genomic region. Each row represents one or more packed reads, with colored segments indicating methylation probability at each position. Reads are packed vertically to minimize plot height while avoiding overlaps.

Value

a ggplot object of the heatmap.

a ggplot plot containing the heatmap.

Examples

```
nmr <- load_example_nanomethresult()
plot_region_heatmap(nmr, "chr7", 6703892, 6730431)
```

plot_violin *Plot violin for regions*

Description

This function plots a violin plot of the methylation proportion for each region in the regions table. The methylation proportion is calculated as the mean of the modification probability within each region, and the violin shows the distribution across groups. Regions are grouped and coloured by the `group_col` column in the `regions` table or `samples(x)`.

Usage

```
plot_violin(
  x,
  regions,
  binary_threshold = 0.5,
  group_col = "group",
  palette = ggplot2::scale_colour_brewer(palette = "Set1")
)
```

Arguments

`x`	the NanoMethResult object.
`regions`	a table of regions containing at least columns chr, strand, start and end. Any additional columns can be used for grouping.
`binary_threshold`	the modification probability such that calls with modification probability above the threshold are considered methylated, and those with probability equal or below are considered unmethylated.
`group_col`	the column to group aggregated trends by. This column can be in from the regions table or samples(x).
`palette`	the ggplot colour palette used for groups.

Value

a ggplot object containing the methylation violin plot.

Examples

```
nmr <- load_example_nanomethresult()
gene_anno <- exons_to_genes(NanoMethViz::exons(nmr))
plot_violin(nmr, gene_anno)
plot_violin(nmr, gene_anno, group_col = "sample")
```

query_exons *Query exons*

Description

Query a data.frame, NanoMethResult or ModBamResult for exon annotation.

Usage

```
query_exons_region(x, chr, start, end)

query_exons_gene_id(x, gene_id)

query_exons_symbol(x, symbol)
```

Arguments

x	the object to query.
chr	the chromosome to query.
start	the start of the query region.
end	the end of the query region.
gene_id	the gene_id to query.
symbol	the gene_id to query.

Value

data.frame of queried exons.

Functions

- query_exons_region(): Query region.
- query_exons_gene_id(): Query gene ID.
- query_exons_symbol(): Query gene symbol.

query_methy *Query methylation data*

Description

Query methylation data

Usage

```
query_methy(
  x,
  chr,
  start,
  end,
  simplify = TRUE,
  force = FALSE,
  truncate = TRUE,
  site_filter = getOption("NanoMethViz.site_filter", 3L)
)
```

Arguments

x	the NanoMethResult object or a path to the methylation data (tabix-bgzipped).
chr	the vector of chromosomes
start	the vector of start positions
end	the vector of end positions
simplify	whether returned results should be row-concatenated
force	whether to force empty output when query region 'chr' does not appear in data. Without 'force', an empty result indicates that the requested 'chr' appears in the data but no data overlaps with requested region, and an invalid 'chr' will cause an error.
truncate	when querying from ModBamFiles, whether or not to truncate returned results to only those within the specified region. Otherwise methylation data for entire reads overlapping the region will be returned.
site_filter	the minimum amount of coverage to report a site. This filters the queried data such that any site with less than the filter is not returned. The default is 1, which means that all sites are returned. This option can be set globally using the `options(NanoMethViz.site_filter = ...)` which will affect all plotting functions in NanoMethViz.

Details

The argument `site_filter` can be set globally using the `options(NanoMethViz.site_filter = ...)` command. The same data entry may appear multiple times in the output if it overlaps multiple regions.

Value

A table containing the data within the queried regions. If simplify is TRUE (default) then returns all data in a single table, otherwise returns a list of tables where each table is the data from one region.

Examples

```
nmr <- load_example_nanomethresult()
query_methy(methy(nmr), "chr7", 6703892, 6730431)
```

---

raw_methy_to_tabix *Convert methylation file to tabix format*

Description

Convert methylation file to tabix format

Usage

```
raw_methy_to_tabix(x)
```

Arguments

x	the path to the sorted methylation file

Value

invisibly returns the path to the tabix file

---

reexports *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

e1071 sigmoid

region_methy_stats *Calculate region methylation statistics*

Description

Calculate the average methylation probability and prevalence based on specified probability threshold.

Usage

```
region_methy_stats(nmr, regions, threshold = 0.5)
```

Arguments

nmr	the NanoMethResult object.
regions	the table of regions to query statistics for.
threshold	the threshold to use for determining methylation calls for the calculation of prevalence.

Value

table of regions with additional columns of methylation summary statistics.

Examples

```
nmr <- load_example_nanomethresult()
gene_anno <- exons_to_genes(NanoMethViz::exons(nmr))
region_methy_stats(nmr, gene_anno)
```

samples *Get sample annotation*

Description

Get sample annotation

Usage

```
samples(object)
```

samples<- *Set sample annotation*

Description

Set sample annotation

Usage

```
samples(object) <- value
```

sort_methy_file *Sort methylation file*

Description

Sort methylation file

Usage

```
sort_methy_file(x)
```

Arguments

x	the path to the methylation file to sort

Value

invisibly returns path of sorted file

Index

∗ **internal**
  assert_valid_exons, 4
  assert_valid_genomic_coords, 4
  assert_valid_methy_samples, 5
  assert_valid_samples, 5
  convert_methy_format, 8
  exons, 10
  exons<-, 10
  get_cgi, 12
  methy, 16
  methy<-, 16
  mod_code, 22
  mod_code<-, 22
  NanoMethViz-package, 3
  raw_methy_to_tabix, 44
  reexports, 44
  samples, 45
  samples<-, 46
  sort_methy_file, 46

assert_valid_exons, 4
assert_valid_genomic_coords, 4
assert_valid_methy_samples, 5
assert_valid_samples, 5

bsseq_to_edger, 6
bsseq_to_log_methy_ratio, 6

cluster_reads, 7
cluster_regions, 8
convert_methy_format, 8
create_tabix_file, 9

exons, 10
exons,ModBamResult-method
  (ModBamResult-class), 19
exons,NanoMethResult-method
  (NanoMethResult-class), 22
exons<-, 10
exons<-,ModBamResult,data.frame-method
  (ModBamResult-class), 19
exons<-,NanoMethResult,data.frame-method
  (NanoMethResult-class), 22
exons_to_genes, 10

filter_methy, 11

get_cgi, 12
get_cgi_grcm39 *(get_cgi_mm10)*, 12
get_cgi_hg19 *(get_cgi_mm10)*, 12
get_cgi_hg38 *(get_cgi_mm10)*, 12
get_cgi_mm10, 12
get_cgi_t2t *(get_cgi_mm10)*, 12
get_example_exons_mus_musculus, 13
get_exons *(get_cgi_mm10)*, 12
get_exons_grcm39 *(get_cgi_mm10)*, 12
get_exons_hg19 *(get_cgi_mm10)*, 12
get_exons_hg38 *(get_cgi_mm10)*, 12
get_exons_homo_sapiens, 14
get_exons_mm10 *(get_cgi_mm10)*, 12
get_exons_mus_musculus, 14
get_exons_t2t *(get_cgi_mm10)*, 12

load_example_modbamresult, 15
load_example_nanomethresult, 15

methy, 16
methy,ModBamResult-method
  (ModBamResult-class), 19
methy,NanoMethResult-method
  (NanoMethResult-class), 22
methy<-, 16
methy<-,ModBamResult,ModBamFiles-method
  (ModBamResult-class), 19
methy<-,NanoMethResult,ANY-method
  (NanoMethResult-class), 22
methy_col_names, 17
methy_to_bsseq, 17
methy_to_edger, 18
mod_code, 22
mod_code,ModBamResult-method
  (ModBamResult-class), 19

mod_code<-, 22
mod_code<-,ModBamResult,character-method
(ModBamResult-class), 19
modbam_to_tabix, 21
ModBamFiles, 18
ModBamFiles-class, 19
ModBamResult (ModBamResult-class), 19
ModBamResult-class, 19

NanoMethResult (NanoMethResult-class),
22
NanoMethResult-class, 22
NanoMethViz (NanoMethViz-package), 3
NanoMethViz-package, 3

plot_agg_genes, 24
plot_agg_regions, 26
plot_gene, 27
plot_gene(), *3*
plot_gene,ModBamResult,character-method
(plot_gene), 27
plot_gene,NanoMethResult,character-method
(plot_gene), 27
plot_gene_heatmap, 29
plot_gene_heatmap,ModBamResult,character-method
(plot_gene_heatmap), 29
plot_gene_heatmap,NanoMethResult,character-method
(plot_gene_heatmap), 29
plot_grange, 31
plot_grange_heatmap, 32
plot_mds, 33
plot_pca, 34
plot_region, 35
plot_region(), *3*
plot_region,ModBamResult,character,numeric,numeric-method
(plot_region), 35
plot_region,ModBamResult,factor,numeric,numeric-method
(plot_region), 35
plot_region,NanoMethResult,character,numeric,numeric-method
(plot_region), 35
plot_region,NanoMethResult,factor,numeric,numeric-method
(plot_region), 35
plot_region_heatmap, 39
plot_region_heatmap,ModBamResult,character,numeric,numeric-method
(plot_region_heatmap), 39
plot_region_heatmap,ModBamResult,factor,numeric,numeric-method
(plot_region_heatmap), 39
plot_region_heatmap,NanoMethResult,character,numeric,numeric-method
(plot_region_heatmap), 39
plot_region_heatmap,NanoMethResult,factor,numeric,numeric-
(plot_region_heatmap), 39
plot_violin, 41

query_exons, 42
query_exons_gene_id (query_exons), 42
query_exons_region (query_exons), 42
query_exons_symbol (query_exons), 42
query_methy, 43

raw_methy_to_tabix, 44
reexports, 44
region_methy_stats, 45

samples, 45
samples,ModBamResult-method
(ModBamResult-class), 19
samples,NanoMethResult-method
(NanoMethResult-class), 22
samples<-, 46
samples<-,ModBamResult,data.frame-method
(ModBamResult-class), 19
samples<-,NanoMethResult,data.frame-method
(NanoMethResult-class), 22
show,ModBamFiles-method (ModBamFiles),
18
sigmoid, *44*
sigmoid (reexports), 44
sort_methy_file, 46