Package ‘NanoMethViz’

May 7, 2024

**Type**  Package

**Title**  Visualise methylation data from Oxford Nanopore sequencing

**Version**  3.0.1

**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

**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, data.table, dbscan, e1071, fs, GenomicRanges, Biostrings, ggrastr, glue, graphics, IRanges, limma (>= 3.44.0), patchwork, purrr, rlang, R.utils, Rsamtools, scales (>= 1.2.0), scico, stats, stringr, tibble, tidyr, utils, withr, zlibbioc

**Suggests**  BiocStyle, DSS, 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
## Contents

NanoMethViz-package .................................................. 3
bsseq_to_edger ......................................................... 4
bsseq_to_log_methy_ratio ........................................... 4
cluster_reads .......................................................... 5
cluster_regions ......................................................... 6
convert_methy_format .................................................. 7
create_tabix_file ....................................................... 7
exons ................................................................. 8
exons<- ............................................................. 8
exons_to_genes ......................................................... 9
filter_methy ............................................................ 9
get_example_exons_mus_musculus ................................. 10
get_exons ............................................................ 10
get_exons_homo_sapiens ............................................ 11
get_exons_mus_musculus ............................................. 12
load_example_modbamresult ......................................... 12
load_example_nanomethresult ..................................... 13
methy ............................................................... 13
methy<- ........................................................... 14
methy_col_names ....................................................... 14
methy_to_bsseq ........................................................ 14
methy_to_edger ........................................................ 15
ModBamFiles .......................................................... 16
ModBamFiles-class .................................................... 16
ModBamResult-class .................................................... 17
modbam_to_tabix .................................................... 19
mod_code ............................................................ 20
mod_code<- ........................................................... 20
NanoMethViz-package

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>
bsseq_to_log_methy_ratio

bsseq_to_log_methy_ratio

bsseq_to_log_methy_ratio

See Also

Useful links:

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

bsseq_to_edger

Convert BSseq object to edgeR methylation matrix

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")
bsseq <- methy_to_bsseq(methy)
edger_mat <- bsseq_to_edger(bsseq)

bsseq_to_log_methy_ratio

Convert BSseq object to log-methylation-ratio matrix

bsseq_to_log_methy_ratio

Description

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

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

```r
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

```r
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**

```r
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**

```r
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"))
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**

```r
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**

```r
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**

```r
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.
get_exons

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

Examples
```r
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_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
```r
get_example_exons_mus_musculus()
```

Value
data.frame containing exons

Examples
```r
eexample_exons <- get_example_exons_mus_musculus()
```

get_exons
Get exon annotations

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

Usage

get_exons_mm10()
get_exons_grcm39()
get_exons_hg19()
get_exons_hg38()

Value
tibble (data.frame) object containing exon annotation.

Examples

mm10_exons <- get_exons_mm10()
grcm39_exons <- get_exons_grcm39()
hg19_exons <- get_exons_hg19()
hg38_exons <- get_exons_hg38()

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

```r
get_exons_mus_musculus()
```

Value

data.frame containing exons

Examples

```r
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

```r
load_example_modbamresult()
```

Value

a ModBamResult object

Examples

```r
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_nanomethresults` without the function call to see how the object is constructed.

**Usage**

```r
load_example_nanomethresult()
```

**Value**

a NanoMethResults object

**Examples**

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

---

**methy**

*Get methylation data*

**Description**

Get methylation data

**Usage**

```r
methy(object)
```

**Arguments**

- `object` the object.

**Value**

the path to the methylation data.

**Examples**

```r
showMethods("methy")
```
**methy**<- \textit{Set methylation data}

**Description**
Set methylation data

**Usage**
methy(object) <- value

**methy_col_names** \textit{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** \textit{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)
methy_to_edger

Arguments

methy the 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(methy, regions = NULL, out_folder = tempdir(), verbose = TRUE)

description

Convert NanoMethResult object to edgeR methylation matrix

Usage

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

Arguments

methy the 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-class

### 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

```r
ModBamFiles(samples, paths)
```

```r
## 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

### 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 a ModBamFiles object, sample information and optional exon information. The object is constructed using the ModBamResult() constructor function described in "Usage".

Usage

```r
## 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.
ModBamResult-class

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 NanoMethResult 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")
),
samples = data.frame(
    sample = "sample1",
    group = "group1"
)
)

modbam_to_tabix(mbr, out_file)
NanoMethResult-class

Description

A NanoMethResult object stores data used for NanoMethViz visualisation. It contains 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)
NanoMethResult-class

## 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

```r
methy <- system.file(package = "NanoMethViz", "methy_subset.tsv.bgz")
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)
```

Description

Plot gene aggregate plot

Usage

```r
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 object.
- `genes`: a character vector of genes to include in aggregate plot, if NULL then all genes 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 to group aggregated trends by. This column can be in from the regions table or samples(x).
- `flank`: the number of flanking bases to add to each side of each region.
- `stranded`: TRUE if negative strand features should have coordinates flipped to reflect features like transcription start sites.
- `span`: the span for loess smoothing.
- `palette`: the ggplot colour palette used for groups.

Value

a ggplot object containing the aggregate methylation trend of genes.

Examples

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

Description

Plot aggregate regions

Usage

```r
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")
)
```
plot_gene

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,
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 ver-
  sion.

Details

This function plots the methylation data for a given gene. 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

```r
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.

Value

- a ggplot object of the heatmap
- a ggplot plot containing the heatmap.
Examples

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

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)
```

Description

Plot GRanges

Usage

```r
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.
 plot_grange_heatmap

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

da 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 gene length.
- **subsample**: the number of read of packed read rows to subsample to.

Value

A ggplot plot containing the heatmap.

Examples

```r
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

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

Arguments

- **x**: the log-methylation-ratio matrix.
- **top**: the number of top genes used to calculate pairwise distances.
- **plot_dims**: the numeric vector of the two dimensions to be plotted.
plot_pca

labels the character vector of labels for data points. By default uses column names of x, set to NULL to plot points.

groups the character vector of groups the data points will be coloured by. Colour palette can be adjusted using scale_colour_*() functions from ggplot2. If groups is numeric, the points will be coloured by a continuous colour palette. By default, groups is NULL and the points will not be coloured.

legend_name the name for the legend.

Value

ggplot object of the MDS plot.

Examples

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

Description

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

Usage

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

Arguments

- **x** the log-methylation-ratio matrix.
- **plot_dims** the numeric vector of the two dimensions to be plotted.
- **labels** the character vector of labels for data points. By default uses column names of x, set to NULL to plot points.
- **groups** the character vector of groups the data points will be coloured by.
- **legend_name** the name for the legend.
Value

ggplot object of the MDS plot.

Examples

```r
nmr <- load_example_nanomethresult()
bss <- methy_to_bsseq(nmr)
llmr <- bsseq_to_log_methy_ratio(bss)
plot_pca(llmr)
```

---

### plot_region

**Plot region methylation**

Plot the methylation of a genomic region.

#### Usage

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

---

#### Description

Plot the methylation of a genomic region.

#### Usage

```r
plot_region(x, chr, start, end, ...)
```
\texttt{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)}

\texttt{## 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)}

\texttt{## 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,
extmap_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 region is specified by chromosome, start and end positions. The basic plot contains a smoothed line plot of the methylation of each experimental group. 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

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

Description

Plot the methylation heatmap of a genomic region.

Usage

```r
plot_region_heatmap(x, chr, start, end, ...)  
```  
```r
## 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 
)
```  
```r
## S4 method for signature 'ModBamResult,character,numeric,numeric'
plot_region_heatmap(
  x, 
  chr, 
  start, 
  end, 
  pos_style = c("to_scale", "compact"),
```
## 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" option 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 gene length.
- **subsample** the number of read of packed read rows to subsample to.

### Value

- a ggplot object of the heatmap.
- a ggplot plot containing the heatmap.
plot_violin

Examples

```r
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 represents the density. The regions are then grouped and coloured by the `group_col` column in the regions table or `samples(x)`.

Usage

```r
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

```r
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**

```r
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", 1L)
)

Arguments

x the NanoMethResults object or a path to the methylation data (tabix-bgzipped).
chr the vector of chromosomes
start the vector of start positions
dead 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(site_filter = ...) which will affect all plotting functions in NanoMethviz.

Details

The argument site_filter can be set globally using the options(site_filter = ...) command.
Value

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

Examples

```r
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

```r
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
  convert_methy_format, 7
  exons, 8
  exons<-, 8
  methy, 13
  methy<-, 14
  mod_code, 20
  mod_code<-, 20
  NanoMethViz-package, 3
  raw_methy_to_tabix, 40
  reexports, 40
  samples, 41
  samples<-, 42
  sort_methy_file, 42
bsseq_to_edger, 4
bsseq_to_log_methy_ratio, 4
cluster_reads, 5
cluster_regions, 6
convert_methy_format, 7
create_tabix_file, 7
exons, 8
exons, ModBamResult-method
  (ModBamResult-class), 17
exons, NanoMethResult-method
  (NanoMethResult-class), 20
exons<-, 8
exons<-, ModBamResult, data.frame-method
  (ModBamResult-class), 17
exons<-, NanoMethResult, data.frame-method
  (NanoMethResult-class), 20
exons_to_genes, 9
filter_methy, 9
get_example_exons_mus_musculus, 10
get_exons, 10
get_exons_grcm39 (get_exons), 10
get_exons_hg19 (get_exons), 10
get_exons_hg38 (get_exons), 10
get_exons_homo_sapiens, 11
get_exons_mm10 (get_exons), 10
get_exons_mus_musculus, 12
load_example_modbamresult, 12
load_example_nanomethresult, 13
methy, 13
methy, ModBamResult-method
  (ModBamResult-class), 17
methy, NanoMethResult-method
  (NanoMethResult-class), 20
methy<-, 14
methy<-, ModBamResult, ModBamFiles-method
  (ModBamResult-class), 17
methy<-, NanoMethResult, ANY-method
  (NanoMethResult-class), 20
methy_col_names, 14
methy_to_bsseq, 14
methy_to_edger, 15
mod_code, 20
mod_code, ModBamResult-method
  (ModBamResult-class), 17
mod_code<-, 20
mod_code<-, ModBamResult, character-method
  (ModBamResult-class), 17
modbam_to_tabix, 19
ModBamFiles, 16
ModBamFiles-class, 16
ModBamResult (ModBamResult-class), 17
ModBamResult-class, 17
NanoMethResult (NanoMethResult-class), 20
NanoMethResult-class, 20
NanoMethViz (NanoMethViz-package), 3
NanoMethViz-package, 3
plot_agg_genes, 22
plot_agg_regions, 23
plot_gene, 24
plot_gene(), 3
plot_gene, ModBamResult, character-method 
  (plot_gene), 24
plot_gene, NanoMethResult, character-method 
  (plot_gene), 24
plot_gene_heatmap, 27
plot_gene_heatmap, ModBamResult, character-method 
  (plot_gene_heatmap), 27
plot_gene_heatmap, NanoMethResult, character-method 
  (plot_gene_heatmap), 27
plot_grange, 28
plot_grange_heatmap, 29
plot_mds, 30
plot_pca, 31
plot_region, 32
plot_region(), 3
plot_region, ModBamResult, character, numeric, numeric-method 
  (plot_region), 32
plot_region, ModBamResult, factor, numeric, numeric-method 
  (plot_region), 32
plot_region, NanoMethResult, character, numeric, numeric-method 
  (plot_region), 32
plot_region, NanoMethResult, factor, numeric, numeric-method 
  (plot_region), 32
plot_region_heatmap, 35
plot_region_heatmap, ModBamResult, character, numeric, numeric-method 
  (plot_region_heatmap), 35
plot_region_heatmap, ModBamResult, factor, numeric, numeric-method 
  (plot_region_heatmap), 35
plot_region_heatmap, NanoMethResult, character, numeric, numeric-method 
  (plot_region_heatmap), 35
plot_region_heatmap, NanoMethResult, factor, numeric, numeric-method 
  (plot_region_heatmap), 35
plot_violin, 37
query_exons, 38
query_exons_gene_id (query_exons), 38
query_exons_region (query_exons), 38
query_exons_symbol (query_exons), 38
query_methy, 39
raw_methy_to_tabix, 40
reexports, 40
region_methy_stats, 41
samples, 41