Package ‘NanoMethViz’

March 12, 2024

Type Package

Title Visualise methylation data from Oxford Nanopore sequencing

Version 2.8.1

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, Biostats, ggrastr, glue, graphics, IRanges, limma (>= 3.44.0), patchwork, purrr, rlang, R.utils, Rsamtools, scales (>= 1.2.0), scico, stats, stringr, tidier, utils, w提示, 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.Mmu.musculus.UCSC.mm10.knownGene, TxDb.Mmu.musculus.UCSC.mm39.refGene, knitr, rmarkdown, rtracklayer, testthat (>= 3.0.0), covr

LinkingTo Rcpp

License Apache License (>= 2.0)

SystemRequirements C++17

VignetteBuilder knitr
### Encoding
UTF-8

### Roxygen
list(markdown = TRUE)

### RoxygenNote
7.2.3

### Config/testthat/parallel
true

### Config/testthat/edition
3

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

**git_branch** RELEASE_3_18

**git_last_commit** 1d8a149

**git_last_commit_date** 2023-11-07

### Repository
Bioconductor 3.18

### Date/Publication
2024-03-11

### Author
Shian Su [cre, aut]

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

### R topics documented:

```
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_nanomethresult ....................................... 12
methy ................................................................. 13
methy<- .............................................................. 13
methy_col_names ....................................................... 14
methy_to_bsseq ........................................................ 14
methy_to_edger ........................................................ 15
ModBamFiles .......................................................... 15
ModBamFiles-class .................................................... 16
ModBamResult-class ................................................... 16
modbam_to_tabix ........................................................ 18
mod_code .............................................................. 19
mod_code<- ........................................................... 19
NanoMethResult-class .................................................. 19
```
NanoMethViz-package

plot_agg_genes .................................................. 21
plot_agg_regions ................................................. 23
plot_gene .......................................................... 24
plot_gene_heatmap ................................................ 26
plot_grange ........................................................ 27
plot_grange_heatmap ............................................. 28
plot_mds ........................................................... 29
plot_pca ............................................................ 30
plot_region ......................................................... 31
plot_region_heatmap ............................................. 33
query_exons ......................................................... 35
query_methy ......................................................... 36
raw_methy_to_tabix ............................................... 37
reexports .......................................................... 37
region_methy_stats ............................................... 38
samples ............................................................ 38
samples< .......................................................... 39
sort_methy_file ..................................................... 39

Index 40

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("ImportingData", package = "NanoMethViz") for how to import data from Nanopolish and f5c.
- See vignette("Introduction", package = "NanoMethViz") for how to create visualisations using this package.

Author(s)

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

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")
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.
cluster_reads

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).
cluster_regions

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

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"))
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

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

get_example_exons_mus_musculus()

Value

data.frame containing exons

Examples

example_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.
**Description**

Get exon annotations for Homo sapiens (hg19)

**Usage**

```r
get_exons_homo_sapiens()
```

**Value**

data.frame containing exons

**Examples**

```r
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_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
load_example_nanomethresult()

Value
a NanoMethResults object

Examples
nmr <- load_example_nanomethresult()
methyl

| methyl | Get methylation data |

**Description**

Get methylation data

**Usage**

methyl(object)

**Arguments**

object the object.

**Value**

the path to the methylation data.

**Examples**

showMethods("methyl")

---

methyl<- Set methylation data

**Description**

Set methylation data

**Usage**

methyl(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 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**

```r
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 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**

```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'
```
ModBamResult-class

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 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
modbam_to_tabix

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.

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

code example

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"
  )
)

Convert BAM with modifications to tabix format
mod_code

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".
NanoMethResult-class

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

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.
Examples

```r
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**

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

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

## S4 method for signature 'ModBamResult,character'
```r
plot_gene(  
  x,  
  gene,  
  window_prop = 0.3,  
  anno_regions = NULL,  
  binary_threshold = NULL,  
  avg_method = c("mean", "median"),  
  spaghetti = FALSE,  
  heatmap = FALSE,  
)
plot_gene

```r
heatmap_subsample = 50, 
span = NULL, 
gene_anno = TRUE, 
palette = ggplot2::scale_colour_brewer(palette = "Set1"), 
line_size = 1, 
mod_scale = c(0, 1)
```

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.
- **span**: the span for loess smoothing.
- **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.

Value

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

Functions

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

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.
plot_grange

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

Description

Plot GRanges

Usage

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

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.

span the span for loess smoothing.

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.

Value

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

Examples

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

Description

Plot GRanges heatmap

Usage

```r
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.
- **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)
mlr <- bsseq_to_log_methy_ratio(bss)
plot_mds(lmr)
```

---

**plot_pca**

**Plot PCA**

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)
mlr <- 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 = FALSE,
  heatmap_subsample = 50,
  span = NULL,
  window_prop = 0,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  mod_scale = c(0, 1)
)

## 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 = FALSE,
  heatmap_subsample = 50,
  span = NULL,
  window_prop = 0,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  mod_scale = c(0, 1)
)
## 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 = FALSE,
  heatmap_subsample = 50,
  span = NULL,
  window_prop = 0,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  mod_scale = c(0, 1)
)

## 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 = FALSE,
  heatmap_subsample = 50,
  span = NULL,
  window_prop = 0,
  palette = ggplot2::scale_colour_brewer(palette = "Set1"),
  line_size = 1,
  mod_scale = c(0, 1)
)

### 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.
plot_region_heatmap

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 gen-
    erated 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.

span the span for loess smoothing.
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.

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 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.

Examples

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

query_exons

<table>
<thead>
<tr>
<th>Query exons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query a data.frame of exons for a subset.</td>
</tr>
</tbody>
</table>

Usage

- `query_exons_region(exons, chr, start, end)`
- `query_exons_gene_id(exons, gene_id)`
- `query_exons_symbol(exons, symbol)`

Arguments

- `exons`: the data.frame of exons.
- `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.
query_methy

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", 1)
)

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
- 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.
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**
```
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**

```r
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**

```r
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**

```r
samples(object)
```
samples<-

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<-, 13
  mod_code, 19
  mod_code<-, 19
  NanoMethViz-package, 3
  raw_methy_to_tabix, 37
  reexports, 37
  samples, 38
  samples<-, 39
  sort_methy_file, 39
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), 16
exons, NanoMethResult-method
  (NanoMethResult-class), 19
exons<-, 8
exons<-, ModBamResult, data.frame-method
  (ModBamResult-class), 16
exons<-, NanoMethResult, data.frame-method
  (NanoMethResult-class), 19
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_nanomethresult, 12
methy, 13
methy, ModBamResult-method
  (ModBamResult-class), 16
methy, NanoMethResult-method
  (NanoMethResult-class), 19
methy<-, 13
methy<-, ModBamResult, ModBamFiles-method
  (ModBamResult-class), 16
methy<-, NanoMethResult, ANY-method
  (NanoMethResult-class), 19
methy_col_names, 14
methy_to_bsseq, 14
methy_to_edger, 15
mod_code, 19
mod_code, ModBamResult-method
  (ModBamResult-class), 16
mod_code<-, 19
mod_code<-, ModBamResult, character-method
  (ModBamResult-class), 16
modbam_to_tabix, 18
ModBamFiles, 15
ModBamFiles-class, 16
ModBamResult (ModBamResult-class), 16
ModBamResult-class, 16
NanoMethResult (NanoMethResult-class), 19
NanoMethResult-class, 19
NanoMethViz (NanoMethViz-package), 3
NanoMethViz-package, 3
plot_agg_genes, 21
plot_agg_regions, 23
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>plot_gene</code></td>
<td>24</td>
</tr>
<tr>
<td><code>plot_gene()</code></td>
<td>3</td>
</tr>
<tr>
<td><code>plot_gene(ModBamResult,character-method)</code></td>
<td>24</td>
</tr>
<tr>
<td><code>plot_gene(NanoMethResult,character-method)</code></td>
<td>24</td>
</tr>
<tr>
<td><code>plot_gene_heatmap</code></td>
<td>26</td>
</tr>
<tr>
<td><code>plot_gene_heatmap(ModBamResult,character-method)</code></td>
<td>26</td>
</tr>
<tr>
<td><code>plot_grange</code></td>
<td>27</td>
</tr>
<tr>
<td><code>plot_mds</code></td>
<td>29</td>
</tr>
<tr>
<td><code>plot_pca</code></td>
<td>30</td>
</tr>
<tr>
<td><code>plot_region</code></td>
<td>31</td>
</tr>
<tr>
<td><code>plot_region()</code></td>
<td>3</td>
</tr>
<tr>
<td><code>plot_region(ModBamResult,character,numeric,numeric-method)</code></td>
<td>31</td>
</tr>
<tr>
<td><code>plot_region(ModBamResult,factor,numeric,numeric-method)</code></td>
<td>31</td>
</tr>
<tr>
<td><code>plot_region(NanoMethResult,character,numeric,numeric-method)</code></td>
<td>31</td>
</tr>
<tr>
<td><code>plot_region(NanoMethResult,factor,numeric,numeric-method)</code></td>
<td>31</td>
</tr>
<tr>
<td><code>plot_region_heatmap</code></td>
<td>33</td>
</tr>
<tr>
<td><code>plot_region_heatmap(ModBamResult,character,numeric,numeric-method)</code></td>
<td>33</td>
</tr>
<tr>
<td><code>plot_region_heatmap(ModBamResult,factor,numeric,numeric-method)</code></td>
<td>33</td>
</tr>
<tr>
<td><code>plot_region_heatmap(NanoMethResult,character,numeric,numeric-method)</code></td>
<td>33</td>
</tr>
<tr>
<td><code>plot_region_heatmap(NanoMethResult,factor,numeric,numeric-method)</code></td>
<td>33</td>
</tr>
<tr>
<td><code>query_exons</code></td>
<td>35</td>
</tr>
<tr>
<td><code>query_exons_gene_id</code></td>
<td>35</td>
</tr>
<tr>
<td><code>query_exons_region</code></td>
<td>35</td>
</tr>
<tr>
<td><code>query_exons_symbol</code></td>
<td>35</td>
</tr>
<tr>
<td><code>query_methy</code></td>
<td>36</td>
</tr>
<tr>
<td><code>raw_methy_to_tabix</code></td>
<td>37</td>
</tr>
<tr>
<td><code>reexports</code></td>
<td>37</td>
</tr>
<tr>
<td><code>region_methy_stats</code></td>
<td>38</td>
</tr>
<tr>
<td><code>samples</code></td>
<td>38</td>
</tr>
<tr>
<td><code>samples,ModBamResult-method</code></td>
<td>(ModBamResult-class), 16</td>
</tr>
<tr>
<td><code>samples&lt;-,ModBamResult,data.frame-method</code></td>
<td>(ModBamResult-class), 16</td>
</tr>
<tr>
<td><code>samples&lt;-,NanoMethResult,data.frame-method</code></td>
<td>(NanoMethResult-class), 19</td>
</tr>
<tr>
<td><code>samples&lt;-,NanoMethResult-class</code>, 19</td>
<td></td>
</tr>
<tr>
<td><code>sigmoid</code></td>
<td>37</td>
</tr>
<tr>
<td><code>sigmoid(reexports)</code></td>
<td>37</td>
</tr>
<tr>
<td><code>sort_methy_file</code></td>
<td>39</td>
</tr>
</tbody>
</table>