Package ‘methrix’

January 23, 2024

Title  Fast and efficient summarization of generic bedGraph files from Bisulfite sequencing

Version  1.16.0

Description  Bedgraph files generated by Bisulfite pipelines often come in various flavors. Critical downstream step requires summarization of these files into methylation/coverage matrices. This step of data aggregation is done by Methrix, including many other useful downstream functions.

License  MIT + file LICENSE

Encoding  UTF-8

LazyData  false

Depends  R (>= 3.6), data.table (>= 1.12.4), SummarizedExperiment

Imports  rtracklayer, DelayedArray, HDF5Array, BSgenome, DelayedMatrixStats, parallel, methods, ggplot2, S4Vectors, matrixStats, graphics, stats, utils, GenomicRanges, IRanges

RoxygenNote  7.1.1

Suggests  knitr, rmarkdown, DSS, hsseq, plotly, BSgenome.Mmuseulus.UCSC.mm9, MafDb.1Kgenomes.phase3.GRCh38, MafDb.1Kgenomes.phase3.hs37d5, BSgenome.Hsapiens.UCSC.hg19, GenomicScores, Biostrings, RColorBrewer, GenomeInfoDb, testthat (>= 2.1.0)

VignetteBuilder  knitr

biocViews  DNAMethylation, Sequencing, Coverage

URL  https://github.com/CompEpigen/methrix

BugReports  https://github.com/CompEpigen/methrix/issues

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

git_branch  RELEASE_3_18

git_last_commit  6ed5233

git_last_commit_date  2023-10-24

Repository  Bioconductor 3.18

Date/Publication  2024-01-23
Author  Anand Mayakonda [aut, cre] (<https://orcid.org/0000-0003-1162-687X>),
Reka Toth [aut] (<https://orcid.org/0000-0002-6096-1052>),
Rajbir Batra [ctb],
Clarissa Feuerstein-Akgöz [ctb],
Joschka Hey [ctb],
Maximilian Schönung [ctb],
Pavlo Lutsik [ctb]

Maintainer  Anand Mayakonda <anand_mt@hotmail.com>

R topics documented:

- combine_methrix .................................................. 3
- convert_HDF5_methrix ........................................... 3
- convert_methrix ................................................... 4
- coverage_filter .................................................... 5
- extract_CPGs ........................................................ 6
- get_matrix .......................................................... 6
- get_region_summary ................................................ 7
- get_stats ............................................................. 8
- load_HDF5_methrix ................................................ 9
- mask_methrix ........................................................ 10
- methrix-class ....................................................... 11
- methrix2bsseq ...................................................... 11
- methrix_data ........................................................ 12
- methrix_pca .......................................................... 12
- methrix_report ...................................................... 13
- order_by_sd .......................................................... 14
- plot_coverage ....................................................... 15
- plot_density ........................................................ 16
- plot_pca ............................................................... 17
- plot_stats ............................................................ 18
- plot_violin ........................................................... 19
- read_bedgraphs ..................................................... 20
- region_filter ........................................................ 22
- remove_snps .......................................................... 23
- remove_uncovered ................................................ 24
- save_HDF5_methrix ............................................... 25
- subset_methrix ...................................................... 25
- write_bedgraphs ................................................... 26
- write_bigwigs ...................................................... 28

Index  29
**combine_methrix**

*Combine methrix objects*

**Description**

Combine methrix objects

**Usage**

combine_methrix(m1, m2, by = c("row", "col"))

**Arguments**

- **m1** Frist methrix object
- **m2** Second methrix object
- **by** The direction of combine. ‘column’ (cbind) combines samples with same regions, ‘row’ combines different regions, e.g. different chromosomes.

**Details**

Takes two methrix objects and combines them row- or column-wise

**Value**

An object of class methrix

---

**convert_HDF5_methrix**

Converts HDF5 methrix object to standard in-memory object.

**Description**

Converts HDF5 methrix object to standard in-memory object.

**Usage**

convert_HDF5_methrix(m = NULL)

**Arguments**

- **m** An object of class methrix, HDF5 format

**Details**

Takes a methrix object and returns with the same object with in-memory assay slots.
### convert_methrix

**Value**

An object of class *methrix*

**Examples**

```r
data(methrix_data)
m2 <- convert_methrix(m=methrix_data)
m <- convert_HDF5_methrix(m=m2)
```

---

**Description**

Converts an in-memory object to an on-disk HDF5 object.

**Usage**

```r
convert_methrix(m = NULL)
```

**Arguments**

- `m` An object of class *methrix*

**Details**

Takes a *methrix* object and returns with the same object with delayed array assay slots with HDF5 backend. Might take long time!

**Value**

An object of class *methrix*, HDF5 format

**Examples**

```r
data(methrix_data)
m2 <- convert_methrix(m=methrix_data)
```
coverage_filter

Filter matrices by coverage

**Usage**

```
coverage_filter(
  m,
  cov_thr = 1,
  min_samples = 1,
  prop_samples = 0,
  group = NULL,
  n_chunks = 1,
  n_cores = 1
)
```

**Arguments**

- `m` *methrix object*
- `cov_thr` minimum coverage required to call a loci covered
- `min_samples` Minimum number of samples that should have a loci with coverage \( \geq \text{cov.thr} \). If `group` is given, then this applies per group. Only need one of `prop_samples` or `min_samples`.
- `prop_samples` Minimum proportion of samples that should have a loci with coverage \( \geq \text{cov.thr} \). If `group` is given, then this applies per group. Only need one of `prop_samples` or `min_samples`.
- `group` a column name from sample annotation that defines groups. In this case, the number of min_samples will be tested group-wise.
- `n_chunks` Number of chunks to split the `methrix` object in case it is very large. Default = 1.
- `n_cores` Number of parallel instances. `n_cores` should be less than or equal to `n_chunks`. If `n_chunks` is not specified, then `n_chunks` is initialized to be equal to `n_cores`. Default = 1.

**Details**

Takes `methrix` object and filters CpGs based on coverage statistics

**Value**

An object of class `methrix`
Examples

data('methrix_data')
# Keep only CpGs which are covered by at least 1 read across 3 samples
coverage_filter(m = methrix_data, cov_thr = 1, min_samples = 3)

extract_CPGs Extracts all CpGs from a genome

Description

Extracts all CpGs from a genome

Usage

exttract_CPGs(ref_genome = NULL)

Arguments

ref_genome BSgenome object or name of the installed BSgenome package. Example: BSgenome.Hsapiens.UCSC.hg19

Value

a list of data.table containing number of CpG's and contig lengths

Examples

## Not run:
hg19_cpgs = methrix::extract_CPGs(ref_genome = 'BSgenome.Hsapiens.UCSC.hg19')

## End(Not run)

get_matrix Extract methylation or coverage matrices

Description

Extract methylation or coverage matrices

Usage

get_matrix(m, type = "M", add_loci = FALSE, in_granges = FALSE)
get_region_summary

Arguments

- **m**: `methrix` object
- **type**: can be `M` or `C`. Default 'M'
- **add_loci**: Default `FALSE`. If `TRUE` adds CpG position info to the matrix and returns as a `data.table`
- **in_granges**: Do you want the outcome in `GRanges`?

Details

Takes `methrix` object and returns user specified methylation or coverage matrix

Value

Coverage or Methylation matrix

Examples

```r
data('methrix_data')
#get methylation matrix
get_matrix(m = methrix_data, type = 'M')
#get methylation matrix along with loci
get_matrix(m = methrix_data, type = 'M', add_loci = TRUE)
#get methylation data as a GRanges object
get_matrix(m = methrix_data, type = 'M', add_loci = TRUE, in_granges = TRUE)
```

get_region_summary

**Extract and summarize methylation or coverage info by regions of interest**

Description

Extract and summarize methylation or coverage info by regions of interest

Usage

```r
get_region_summary(
  m,
  regions = NULL,
  type = "M",
  how = "mean",
  overlap_type = "within",
  na.rm = TRUE,
  elementMetadata.col = NULL,
  verbose = TRUE,
  n_chunks = 1,
  n_cores = 1
)
```
get_stats

Estimate descriptive statistics

description

Estimate descriptive statistics

Usage

get_stats(m, per_chr = TRUE)


### load_HDF5_methrix

**Description**

Loads HDF5 methrix object

**Usage**

```r
load_HDF5_methrix(dir = NULL, ...)
```

**Arguments**

- `dir` The directory to read in from. Default NULL
- `...` Parameters to pass to loadHDF5SummarizedExperiment

**Details**

Takes directory with a previously saved HDF5Array format `methrix` object and loads it

**Value**

An object of class `methrix`

---

### get_stats

**Arguments**

- `methrix` object
- `per_chr` Estimate stats per chromosome. Default TRUE

**Details**

Calculate descriptive statistics

**Value**

data.table of summary stats

**See Also**

`plot_stats`

**Examples**

```r
data('methrix_data')
get_stats(methrix_data)
```
Examples

data('methrix_data')
methrix_data_h5 <- convert_methrix(m=methrix_data)
target_dir = paste0(getwd(), '/temp1/')
save_HDF5_methrix(methrix_data_h5, dir = target_dir, replace = TRUE)
load_HDF5_methrix(target_dir)

mask_methrix  

Masks too high or too low coverage

Description

Masks too high or too low coverage

Usage

mask_methrix(m, low_count = NULL, high_quantile = 0.99, n_cores = 1)

Arguments

m  methrix object
low_count  The minimal coverage allowed. Everything below, will get masked. Default = NULL, nothing gets masked.
high_quantile  The quantile limit of coverage. Quantiles are calculated for each sample and everything that belongs to a higher quantile than the defined will be masked. Default = 0.99.
n_cores  Number of parallel instances. Can only be used if methrix is in HDF5 format. Default = 1.

Details

Takes methrix object and masks sites with too high or too low coverage by putting NA for coverage and beta value. The sites will remain in the object.

Value

An object of class methrix

Examples

data('methrix_data')
mask_methrix(m = methrix_data, low_count = 5, high_quantile = 0.99 )
Class methrix

Slots

- **assays**: A list of two matrices containing 'Methylation' and 'Coverage' information.
- **elementMetadata**: A DataFrame describing rows in corresponding assay matrices.
- **colData**: genome: the name of the BSgenome that was used to extract CpGs, isHDF5: is it stored in HDF5 Array format.
- **metadata**: a list of meta data associated with the assays.
- **NAMES**: NULL.

Convert methrix to bsseq object

Description

Convert methrix to bsseq object

Usage

methrix2bsseq(m)

Arguments

m

methrix object

Details

Takes methrix object and returns a bsseq object

Value

An object of class bsseq

Examples

```r
## Not run:
data('methrix_data')
methrix2bsseq(m = methrix_data)
## End(Not run)
```
**methrix_data**  
*WGBS for colon cancer, chr21 and chr22*

**Description**

This is a subset of original ‘bsseqData’ converted to ‘methrix’ containing Whole-genome bisulfite sequencing data (WGBS) for colon cancer on chromosome 21 and 22.

**Usage**

```r
data('methrix_data')
```

**Format**

An object of class 'methrix'

**References**


**Examples**

```r
data('methrix_data')
methrix_data
```

**methrix_pca**  
*Principal Component Analysis*

**Description**

Principal Component Analysis

**Usage**

```r
methrix_pca(
m,
var = "top",
top_var = 1000,
ranges = NULL,
pheno = NULL,
do_plot = TRUE,
n_pc = 2
)
```
**methrix_report**

**Arguments**

- **m**
  - Input `methrix` object

- **var**
  - Choose between random CpG sites ('rand') or most variable CpGs ('top').

- **top_var**
  - Number of variable CpGs to use. Default 1000 Set it to NULL to use all CpGs (which is not recommended due to memory requirements). This option is mutually exclusive with `ranges`.

- **ranges**
  - Genomic regions to be summarized. Could be a data.table with 3 columns (chr, start, end) or a GenomicRanges object.

- **pheno**
  - Column name of colData(m). Default NULL. Will be used as a factor to color different groups.

- **do_plot**
  - Should a plot be generated?

- **n_pc**
  - Default 2.

**Value**

- PCA results

**Examples**

```r
data('methrix_data')
methrix_pca(methrix_data, do_plot = FALSE)
```

---

**methrix_report**  
*Creates a detailed interactive html summary report from Methrix object*

**Description**

Creates a detailed interactive html summary report from Methrix object. If the directory contains required files (from previous run), it directly proceeds to generate html report.

**Usage**

```r
methrix_report(
  meth,
  output_dir = NULL,
  recal_stats = FALSE,
  plot_beta_dist = TRUE,
  beta_nCpG = 10000,
  prefix = NULL,
  n_thr = 4
)
```
order_by_sd

Arguments

meth  methrix object
output_dir Output directory name where the files should be saved. If NULL creates a tempdir
recal_stats Whether summary statistics should be recalculated? If you are using subsetted methrix object set this to TRUE.
plot_beta_dist Default TRUE. Can be time consuming.
beta_nCpG Number of CpGs rto use for estimating beta value distribution. Default 10000
prefix If provided, the name of the report and the intermediate files will start with the prefix.
n_thr Default 4. Only used if plot_beta_dist is TRUE

Value

an interactive html report

Examples

## Not run:
data('methrix_data')
methrix::methrix_report(meth = methrix_data)
## End(Not run)

order_by_sd  Order methrix object by SD

Description

Order methrix object by SD

Usage

order_by_sd(m)

Arguments

m  methrix object

Details

Takes methrix object and reorganizes the data by standard deviation

Value

An object of class methrix
plot_coverage

Examples

data('methrix_data')
order_by_sd(m = methrix_data)

plot_coverage

Description

Coverage QC Plots

Usage

plot_coverage(
m, type = c("hist", "dens"), pheno = NULL, perGroup = FALSE, lim = 100, size.lim = 1e+06, col_palatte = "RdYlGn"
)

Arguments

m Input methrix object

type Choose between 'hist' (histogram) or 'dens' (density plot).

pheno Column name of colData(m). Will be used as a factor to color different groups in the plot.

perGroup Color the plots in a sample-wise manner?

lim Maximum coverage value to be plotted.

size.lim The maximum number of observations (sites*samples) to use. If the dataset is larger that this, random sites will be selected from the genome.

col_palette Name of the RColorBrewer palette to use for plotting.

Value

ggplot2 object

Examples

data('methrix_data')
plot_coverage(m = methrix_data)
Description

Density Plot of β-Values

Usage

```r
plot_density(
m, 
ranges = NULL, 
n_cpgs = 25000, 
pheno = NULL, 
col_palette = "RdYlGn"
)
```

Arguments

- **m**: Input `methrix` object
- **ranges**: genomic regions to be summarized. Could be a data.table with 3 columns (chr, start, end) or a GenomicRanges object
- **n_cpgs**: Use these many random CpGs for plotting. Default 25000. Set it to `NULL` to use all - which can be memory expensive.
- **pheno**: Column name of colData(m). Will be used as a factor to color different groups in the violin plot.
- **col_palette**: Name of the RColorBrewer palette to use for plotting.

Value

`ggplot2` object

Examples

```r
data('methrix_data')
plot_density(m = methrix_data)
```
Description

Plot PCA results

Usage

plot_pca(
  pca_res,
  m = NULL,
  col_anno = NULL,
  shape_anno = NULL,
  pc_x = "PC1",
  pc_y = "PC2",
  show_labels = FALSE
)

Arguments

- pca_res: Results from methrix_pca
- m: Optional methrix object. Default NULL
- col_anno: Column name of colData(m). Default NULL. Will be used as a factor to color different groups. Required methrix object
- shape_anno: Column name of colData(m). Default NULL. Will be used as a factor to shape different groups. Required methrix object
- pc_x: Default 'PC1'
- pc_y: Default 'PC2'
- show_labels: Default FALSE

Value

ggplot2 object

Examples

data('methrix_data')
mpc = methrix_pca(methrix_data, do_plot = FALSE)
plot_pca(mpc)
### plot_stats

#### Plot descriptive statistics

**Description**

Plot descriptive statistics

**Usage**

```r
plot_stats(
  plot_dat,
  what = "M",
  stat = "mean",
  ignore_chr = NULL,
  samples = NULL,
  n_col = NULL,
  n_row = NULL
)
```

**Arguments**

- `plot_dat`: results from `get_stats`
- `what`: Can be M or C. Default M
- `stat`: Can be mean or median. Default mean
- `ignore_chr`: Chromosomes to ignore. Default NULL
- `samples`: Use only these samples. Default NULL
- `n_col`: number of columns. Passed to ‘facet_wrap’
- `n_row`: number of rows. Passed to ‘facet_wrap’

**Details**

plot descriptive statistics results from `get_stats`

**Value**

ggplot2 object

**See Also**

`get_stats`

**Examples**

```r
data('methrix_data')
gs = get_stats(methrix_data)
plot_stats(gs)
```
plot_violin

Violin Plot for $\beta$-Values

Description

Violin Plot for $\beta$-Values

Usage

```r
plot_violin(
  m,
  ranges = NULL,
  n_cpgs = 25000,
  pheno = NULL,
  col_palette = "RdYlGn"
)
```

Arguments

- **m**: Input `methrix` object
- **ranges**: genomic regions to be summarized. Could be a data.table with 3 columns (chr, start, end) or a GenomicRanges object
- **n_cpgs**: Use these many random CpGs for plotting. Default 25000. Set it to NULL to use all - which can be memory expensive.
- **pheno**: Column name of colData(m). Will be used as a factor to color different groups in the violin plot.
- **col_palette**: Name of the RColorBrewer palette to use for plotting.

Value

`ggplot2` object

Examples

```r
data('methrix_data')
plot_violin(m = methrix_data)
```
read_bedgraphs

Versatile BedGraph reader.

Description

Versatile BedGraph reader.

Usage

read_bedgraphs(
  files = NULL,
  pipeline = NULL,
  zero_based = TRUE,
  stranded = FALSE,
  collapse_strands = FALSE,
  ref_cpgs = NULL,
  ref_build = NULL,
  contigs = NULL,
  vect = FALSE,
  vect_batch_size = NULL,
  coldata = NULL,
  chr_idx = NULL,
  start_idx = NULL,
  end_idx = NULL,
  beta_idx = NULL,
  M_idx = NULL,
  U_idx = NULL,
  strand_idx = NULL,
  cov_idx = NULL,
  synced_coordinates = FALSE,
  n_threads = 1,
  h5 = FALSE,
  h5_dir = NULL,
  h5temp = NULL,
  verbose = TRUE
)

Arguments

files  bedgraph files.
pipeline Default NULL. Currently supports "Bismark_cov", "MethylDackel", "MethylcTools", "BisSNP", "BSseeker2_CGmap" If not known use idx arguments for manual column assignments.
zero_based Are bedgraph regions zero based? Default TRUE
stranded Default FALSE
read_bedgraphs

collapse_strands
If TRUE collapses CpGs on different crick strand into watson. Default FALSE

ref_cpgs
BSgenome object, or name of the installed BSgenome package, or an output from extract_CPGs. Example: BSgenome.Hsapiens.UCSC.hg19

ref_build
reference genome for bedgraphs. Default NULL. Only used for additional details. Doesnt affect in any way.

contigs
contigs to restrict genomic CpGs to. Default all autosomes and allosomes - ignoring extra contigs.

vect
To use vectorized code. Default FALSE. Set to TRUE if you don’t have large number of BedGraph files.

vect_batch_size
Default NULL. Process samples in batches. Applicable only when vect = TRUE

coldata
An optional DataFrame describing the samples. Row names, if present, become the column names of the matrix. If NULL, then a DataFrame will be created with basename of files used as the row names.

chr_idx
column index for chromosome in bedgraph files

start_idx
column index for start position in bedgraph files

end_idx
column index for end position in bedgraph files

beta_idx
column index for beta values in bedgraph files

M_idx
column index for read counts supporting Methylation in bedgraph files

U_idx
column index for read counts supporting Un-methylation in bedgraph files

strand_idx
column index for strand information in bedgraph files

cov_idx
column index for total-coverage in bedgraph files

synced_coordinates
Are the start and end coordinates of a stranded bedgraph are synchronized between + and - strands? Possible values: FALSE (default), TRUE if the start coordinates are the start coordinates of the C on the plus strand.

n_threads
number of threads to use. Default 1. Be-careful - there is a linear increase in memory usage with number of threads. This option is does not work with Windows OS.

h5
Should the coverage and methylation matrices be stored as 'HDF5Array'

h5_dir
directory to store H5 based object

h5temp
temporary directory to store hdf5

verbose
Be little chatty ? Default TRUE.

Details
Reads BedGraph files and generates methylation and coverage matrices. Optionally arrays can be serialized as on-disk HDF5 arrays.

Value
An object of class methrix
region_filter

Filter matrices by region

Description
Filter matrices by region

Usage
region_filter(m, regions, type = "within")

Arguments

m methrix object
regions genomic regions to filter-out. Could be a data.table with 3 columns (chr, start, end) or a GenomicRanges object

type defines the type of the overlap of the CpG sites with the target regions. Default value is 'within'. For detailed description, see the foverlaps function of the data.table package.

Details
Takes methrix object and filters CpGs based on supplied regions in data.table or GRanges format

Value
An object of class methrix

Examples
data('methrix_data')
region_filter(m = methrix_data, regions = data.table(chr = 'chr21', start = 27867971, end = 27868103))
remove_snps

Removes CpG sites from the object if they overlap with common SNPs

Description

Removes CpG sites from the object if they overlap with common SNPs

Usage

remove_snps(
  m,
  populations = NULL,
  maf_threshold = 0.01,
  reduce_filtering = FALSE,
  forced = FALSE,
  keep = FALSE,
  n_chunks = 1,
  n_cores = 1
)

Arguments

m methrix object
populations Populations to use. Default is all.
maf_threshold The frequency threshold, above which the SNPs will be removed. Default is 0.01
reduce_filtering If TRUE, the SNPs with a MAF < 0.1 will be evaluated and only the highly variable ones will be removed. Default FALSE.
forced the reduce_filtering is not recommended with less than 10 samples, but can be forced. Default is FALSE.
keep Do you want to keep the sites that were filtered out? In this case, the function will return with a list of wo methrix objects.
n_chunks Number of chunks to split the methrix object in case it is very large. Can only be used if input data is in HDF5 format. Default = 1.
n_cores Number of parallel instances. Can only be used if input data is in HDF5 format. n_cores should be less than or equal to n_chunks. If n_chunks is not specified, then n_chunks is initialized to be equal to n_cores. Default = 1.

Details

Takes methrix object and removes common SNPs. SNPs overlapping with a CpG site and have a minor allele frequency (MAF) above a threshold in any of the populations used will be selected and the corresponding CpG sites will be removed from the methrix object. With the reduce_filtering option, SNPs with MAF < 0.1 will be further evaluated. If they show low variance in the dataset, there is probably no genotype variability in the population, therefore the corresponding CpG site won’t be removed. Please keep in mind that variance thresholds are
Value

methrix object or a list of methrix objects

Examples

data('methrix_data')
remove_snps(m = methrix_data, maf_threshold=0.01)

---

remove_uncovered  Remove loci that are uncovered across all samples

Description

Remove loci that are uncovered across all samples

Usage

remove_uncovered(m)

Arguments

m  methrix object

Details

Takes methrix object and removes loci that are uncovered across all samples

Value

An object of class methrix

Examples

data('methrix_data')
remove_uncovered(m = methrix_data)
save_HDF5_methrix  Saves HDF5 methrix object

Description
Saves HDF5 methrix object

Usage
save_HDF5_methrix(m = NULL, dir = NULL, replace = FALSE, ...)

Arguments
- m  methrix object
- dir The directory to use. Created, if not existing. Default NULL.
- replace Should it overwrite the pre-existing data? FALSE by default.
- ... Parameters to pass to saveHDF5SummarizedExperiment

Details
Takes methrix object and saves it

Value
Nothing

Examples
data('methrix_data')
methrix_data_h5 <- convert_methrix(m=methrix_data)
target_dir = paste0(getwd(), '/temp/')
save_HDF5_methrix(methrix_data_h5, dir = target_dir, replace = TRUE)

subset_methrix  Subsets methrix object based on given conditions.

Description
Subsets methrix object based on given conditions.
Usage

```r
subset_methrix(
  m,
  regions = NULL,
  contigs = NULL,
  samples = NULL,
  overlap_type = "within"
)
```

Arguments

- `m` *methrix* object
- `regions` genomic regions to subset by. Could be a data.table with 3 columns (chr, start, end) or a GenomicRanges object
- `contigs` chromosome names to subset by
- `samples` sample names to subset by
- `overlap_type` defines the type of the overlap of the CpG sites with the target region. Default value is ‘within’. For detailed description, see the `foverlaps` function of the `data.table` package.

Details

Takes *methrix* object and filters CpGs based on coverage statistics

Value

An object of class *methrix*

Examples

```r
data('methrix_data')
#Subset to chromosome 1
subset_methrix(methrix_data, contigs = 'chr21')
```

---

**write_bedgraphs**

**Writes bedGraphs from methrix object**

Description

Writes bedGraphs from methrix object
write_bedgraphs

Usage

write_bedgraphs(
  m,  
  output_dir = NULL,  
  rm_NA = TRUE,  
  force = FALSE,  
  n_thr = 4,  
  compress = TRUE,  
  SeqStyle = "UCSC",  
  multiBed = NULL,  
  metilene = FALSE,  
  phenoCol = NULL,  
  add_coverage = FALSE
)

Arguments

m  methrix object
output_dir  Output directory name where the files should be saved. If NULL creates a tempdir
rm_NA  remove NAs
force  forces to create files if they are existing
n_thr  Default 4.
compress  Whether to compress the output. Default TRUE
SeqStyle  Default 'UCSC' with 'chr' prefix.
multiBed  Default NULL. If provided a filename, a single bedGraph file with all samples included is generated.
metilene  Default FALSE. If TRUE outputs bedgraphs in 'metilene' format that can be directly used for DMR calling with 'metilene'. This option works only when multiBed = TRUE.
phenoCol  Default NULL. 'condition' column from colData. Only applicable if metilene = TRUE
add_coverage  Should the output file contain information on coverage? Default FALSE

Value

writes bedgraph files to output

Examples

data('methrix_data')
write_bedgraphs(m = methrix_data, output_dir = './temp')
# Export to metiline format for DMR calling with metiline
write_bedgraphs(m = methrix_data, output_dir = './temp', rm_NA = FALSE, 
metilene = TRUE, multiBed = "metiline_ip", phenoCol = "Condition")
write_bigwigs  

Exports methrix object as bigWigs

Usage

write_bigwigs(m, output_dir = getwd(), samp_names = NULL)

Arguments

m  
methrix object

output_dir  
Output directory name where the files should be saved. Default getwd()

samp_names  
sample names to export

Examples

data('methrix_data')
write_bigwigs(m = methrix_data, output_dir = './temp')
Index

* datasets
  - methrix_data, 12
  - combine_methrix, 3
  - convert_HDF5_methrix, 3
  - convert_methrix, 4
  - coverage_filter, 5
  - data.table, 22, 26
  - extract_CPGs, 6, 21
  - get_matrix, 6
  - get_region_summary, 7
  - get_stats, 8, 18
  - IRanges, 8
  - load_HDF5_methrix, 9
  - mask_methrix, 10
  - methrix, 3–5, 7–11, 13–16, 19, 21–28
  - methrix (methrix-class), 11
  - methrix-class, 11
  - methrix2bsseq, 11
  - methrix_data, 12
  - methrix_pca, 12, 17
  - methrix_report, 13
  - order_by_sd, 14
  - plot_coverage, 15
  - plot_density, 16
  - plot_pca, 17
  - plot_stats, 9, 18
  - plot_violin, 19
  - read_bedgraphs, 20
  - region_filter, 22
  - remove_snps, 23
  - remove_uncovered, 24
  - save_HDF5_methrix, 25
  - subset_methrix, 25
  - write_bedgraphs, 26
  - write_bigwigs, 28