Package 'methodical'

September 19, 2024

Title Discovering genomic regions where methylation is strongly associated with transcriptional activity

Version 1.1.0

Description
DNA methylation is generally considered to be associated with transcriptional silencing. However, comprehensive, genome-wide investigation of this relationship requires the evaluation of potentially millions of correlation values between the methylation of individual genomic loci and expression of associated transcripts in a relatively large numbers of samples. Methodical makes this process quick and easy while keeping a low memory footprint. It also provides a novel method for identifying regions where a number of methylation sites are consistently strongly associated with transcriptional expression. In addition, Methodical enables housing DNA methylation data from diverse sources (e.g. WGBS, RRBS and methylation arrays) with a
common framework, lifting over DNA methylation data between different genome builds and creating base-resolution plots of the association between DNA methylation and transcriptional activity at transcriptional start sites.

License GPL (>= 3)

BugReports https://github.com/richardheery/methodical/issues

biocViews DNAMethylation, MethylationArray, Transcription, GenomeWideAssociation, Software

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Depends GenomicRanges, ggplot2, R (>= 4.2.0), SummarizedExperiment

LazyData false

Imports BiocParallel, Biostrings, BSgenome, cowplot, data.table, DelayedArray, dplyr, ExperimentHub, foreach, GenomeInfoDb, HDF5Array, IRanges, R.utils, RcppRoll, rhdf5, rtracklayer, S4Vectors, scales, tibble, tidyr

Suggests AnnotationHub, annotatr, BiocStyle, BSgenome.Athaliana.TAIR.TAIR9, BSgenome.Hsapiens.UCSC.hg19, BSgenome.Hsapiens.UCSC.hg38, DESeq2, knitr, methrix, rmarkdown, TumourMethData

VignetteBuilder knitr

SystemRequirements kallisto

URL https://github.com/richardheery/methodical

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

git_branch devel

git_last_commit 3aab960

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-09-18

Author Richard Heery [aut, cre] (<https://orcid.org/0000-0001-8067-3114>)

Maintainer Richard Heery <richardheery@gmail.com>

Contents

`methodical-package` *methodical: A one-stop shop for dealing with big DNA methylation datasets*

Description

DNA methylation is generally considered to be associated with transcriptional silencing. However, comprehensive, genome-wide investigation of this relationship requires the evaluation of potentially millions of correlation values between the methylation of individual genomic loci and expression of associated transcripts in a relatively large numbers of samples. Methodical makes this process quick and easy while keeping a low memory footprint. It also provides a novel method for identifying regions where a number of methylation sites are consistently strongly associated with transcriptional expression. In addition, Methodical enables housing DNA methylation data from diverse sources (e.g. WGBS, RRBS and methylation arrays) with a common framework, lifting over DNA methylation data between different genome builds and creating base-resolution plots of the association between DNA methylation and transcriptional activity at transcriptional start sites.

Author(s)

Richard Heery

See Also

Useful links:

- https://github.com/richardheery/methodical
- Report bugs at https://github.com/richardheery/methodical/issues

.calculate_regions_intersections

Calculate the number of bases in the intersection of two GRanges objects

Description

Calculate the number of bases in the intersection of two GRanges objects

Usage

```
.calculate_regions_intersections(
  gr1,
  gr2,
  ignore.strand = TRUE,
  overlap_measure = "absolute"
)
```

Arguments

gr1	A GRanges object
gr2	A GRanges object
ignore.strand	TRUE or FALSE indicating whether strand should be ignored when calculating intersections. Default is TRUE.
overlap_measure	One of "absolute", "proportion" or "jaccard" indicating whether to calculate the absolute size of the intersection in base pairs, the proportion base pairs of gr1 overlapping gr2 or the Jaccard index of the intersection in terms of base pairs. Default value is "absolute".

Value

An numeric value

.chunk_regions

Split genomic regions into balanced chunks based on the number of methylation sites that they cover

Description

Split genomic regions into balanced chunks based on the number of methylation sites that they cover

Usage

```
.chunk_regions(
  meth_rse,
  genomic_regions,
  max_sites_per_chunk = NULL,
  ncores = 1
)
```

Arguments

meth_rse	A RangedSummarizedExperiment with methylation values.
genomic_regions	A GRanges object.
max_sites_per_chunk	The maximum number of methylation sites to load into memory at once for each chunk.
ncores	The number of cores that will be used.

Value

A GRangesList where each GRanges object overlaps approximately the number of methylation sites given by max_sites_per_chunk

---

.count_covered_bases *Calculate the number of unique bases covered by all regions in a GRanges object*

Description

Calculate the number of unique bases covered by all regions in a GRanges object

Usage

```
.count_covered_bases(gr)
```

Arguments

gr	A GRanges object

Value

An numeric value

.create_meth_rse_from_hdf5 *Create a RangedSummarizedExperiment for methylation values already deposited in HDF5*

Description

Create a RangedSummarizedExperiment for methylation values already deposited in HDF5

Usage

```
.create_meth_rse_from_hdf5(
  hdf5_filepath,
  hdf5_dir,
  meth_sites_df,
  sample_metadata
)
```

Arguments

hdf5_filepath	Path to HDF5 file
hdf5_dir	The path to the HDF5 directory.
meth_sites_df	A data.frame with the positions of methylation sites
sample_metadata	A data.frame with sample metadata

Value

A RangedSummarizedExperiment with methylation values

.make_meth_rse_setup *Perform setup for makeMethRSEFromBedgraphs or makeMethRSEFromArrayFiles*

Description

Perform setup for makeMethRSEFromBedgraphs or makeMethRSEFromArrayFiles

Usage

```
.make_meth_rse_setup(
  meth_files,
  meth_sites,
  sample_metadata,
  hdf5_dir,
  dataset_name,
  overwrite,
  chunkdim,
  temporary_dir,
  ...
)
```

Arguments

meth_files	A vector of paths to files with methylation values. Automatically detects if meth_files contain a header if every field in the first line is a character.
meth_sites	A GRanges object with the locations of the methylation sites of interest. Any regions in meth_files that are not in meth_sites are ignored.
sample_metadata	Sample metadata to be used as colData for the RangedSummarizedExperiment.
hdf5_dir	Directory to save HDF5 file. Is created if it doesn't exist. HDF5 file is called assays.h5.
dataset_name	Name to give data set in HDF5 file.
overwrite	TRUE or FALSE indicating whether to allow overwriting if dataset_name already exists in assays.h5.
chunkdim	The dimensions of the chunks for the HDF5 file.
temporary_dir	Name to give a temporary directory to store intermediate files. A directory with this name cannot already exist.
...	Additional arguments to be passed to HDF5Array::HDF5RealizationSink.

Value

A list describing the setup to be used for makeMethRSEFromBedgraphs or makeMethRSEFromArrayFiles.

`.split_bedgraph` *Split data from a single methylation array files into chunks*

Description

Split data from a single methylation array files into chunks

Usage

```
.split_bedgraph(bg_file, column, file_count, parameters)
```

Arguments

`bg_file`	Path to a bedgraph file.
`column`	The current grid column being processed.
`file_count`	The number of the current file being processed.
`parameters`	A list of parameters for processing the bedgraph.

Value

Invisibly returns NULL.

`.split_bedgraphs_into_chunks` *Split data from bedGraph files into chunks*

Description

Split data from bedGraph files into chunks

Usage

```
.split_bedgraphs_into_chunks(
  bedgraphs,
  seqnames_column,
  start_column,
  end_column,
  value_column,
  file_grid_columns,
  meth_sites,
  meth_site_groups,
  temp_chunk_dirs,
  zero_based,
  normalization_factor,
  decimal_places,
  BPPARAM
)
```

Arguments

bedgraphs	Paths to bedgraph files.
seqnames_column	The column number in bedgraphs which corresponds to the sequence names.
start_column	The column number in bedgraphs which corresponds to the start positions.
end_column	The column number in bedgraphs which corresponds to the end positions.
value_column	The column number in bedgraphs which corresponds to the methylation values.
file_grid_columns	The grid column number for each file.
meth_sites	A GRanges object with the locations of the methylation sites of interest.
meth_site_groups	A list with the indices of the methylation sites in each group.
temp_chunk_dirs	A vector giving the temporary directory associated with each chunk.
zero_based	TRUE or FALSE indicating if files are zero-based.
normalization_factor	An optional numerical value to divide methylation values by to convert them to fractions e.g. 100 if they are percentages. Default is not to leave values as they are in the input files.
decimal_places	Integer indicating the number of decimal places to round beta values to.
BPPARAM	A BiocParallelParam object.

Value

A data.table with the methylation sites sorted by seqnames and start.

.split_meth_array_file *Split data from a single methylation array files into chunks*

Description

Split data from a single methylation array files into chunks

Usage

```
.split_meth_array_file(file, column, file_count, parameters)
```

Arguments

file	Path to a methylation array file.
column	The current grid column being processed.
file_count	The number of the file being processed
parameters	A list of parameters for processing the bedgraph.

Value

Invisibly returns NULL.

`.split_meth_array_files_into_chunks` *Split data from methylation array files into chunks*

Description

Split data from methylation array files into chunks

Usage

```
.split_meth_array_files_into_chunks(
  array_files,
  probe_name_column,
  beta_value_column,
  file_grid_columns,
  probe_ranges,
  probe_groups,
  temp_chunk_dirs,
  normalization_factor,
  decimal_places,
  BPPARAM
)
```

Arguments

`array_files`	Paths to methylation array files.
`probe_name_column`	The column number in array_files which corresponds to the name of the probes. Default is 1st column.
`beta_value_column`	The column number in array_files which corresponds to the beta values. Default is 2nd column.
`file_grid_columns`	The grid column number for each file.
`probe_ranges`	A GRanges object giving the genomic locations of probes where each region corresponds to a separate probe.
`probe_groups`	A list with the indices of the probes in each group.
`temp_chunk_dirs`	A vector giving the temporary directory associated with each chunk.
`normalization_factor`	An optional numerical value to divide methylation values by to convert them to fractions e.g. 100 if they are percentages. Default is not to leave values as they are in the input files.
`decimal_places`	Integer indicating the number of decimal places to round beta values to.
`BPPARAM`	A BiocParallelParam object.

Value

A data.table with the probe sites sorted by seqnames, start and probe name.

.summarize_chunk_methylation *Summarize methylation values for regions in a chunk*

Description

Summarize methylation values for regions in a chunk

Usage

```
.summarize_chunk_methylation(
  chunk_regions,
  meth_rse,
  assay_number,
  summary_function,
  na.rm,
  ...
)
```

Arguments

chunk_regions	Chunk with genomic regions of interest.
meth_rse	A RangedSummarizedExperiment with methylation values.
assay_number	The assay from meth_rse to extract values from.
summary_function	summary_function A function that summarizes column values.
na.rm	TRUE or FALSE indicating whether to remove NA values when calculating summaries.
...	Additional arguments to be passed to summary_function.

Value

A function which returns a list with the

.test_tmrs *Find TMRs where smoothed methodical scores exceed thresholds*

Description

Find TMRs where smoothed methodical scores exceed thresholds

Usage

```
.test_tmrs(
  meth_sites_gr,
  smoothed_methodical_scores,
  p_value_threshold = 0.005,
  tss_gr = NULL,
  transcript_id = NULL
)
```

Arguments

meth_sites_gr	A GRanges object with the location of methylation sites.
smoothed_methodical_scores	A numeric vector with the smoothed methodical scores associated with each methylation site.
p_value_threshold	The p_value cutoff to use. Default value is 0.005.
tss_gr	An optional GRanges object giving the location of the TSS meth_sites_gr is associated with.
transcript_id	Name of the transcript associated with the TSS.

Value

A GRanges object with the location of TMRs.

.tss_correlations *Calculate meth site-transcript correlations for given TSS*

Description

Calculate meth site-transcript correlations for given TSS

Usage

```
.tss_correlations(correlation_objects)
```

Arguments

correlation_objects	A list with a table of methylation values, expression values for transcripts, a GRanges for the TSS and the name of the transcript.

Value

A data.frame with the correlation values

.tss_iterator *Create an iterator function for use with bpiterate*

Description

Create an iterator function for use with bpiterate

Usage

```
.tss_iterator(
  meth_values_chunk,
  tss_region_indices_list,
  transcript_values,
  tss_for_chunk,
  cor_method,
  add_distance_to_region,
  results_dir
)
```

Arguments

meth_values_chunk	A table with methylation values for current chunk
tss_region_indices_list	A list with the indices for methylation sites associated with each TSS.
transcript_values	A list with expression values for transcripts.
tss_for_chunk	A list of GRanges with the TSS for the current chunk.
cor_method	Correlation method to use.
add_distance_to_region	TRUE or FALSE indicating whether to add distance to TSS.
results_dir	Location of results directory.

Value

An iterator function which returns a list with the parameters necessary for .tss_correlations.

.write_chunks_to_hdf5 *Write chunks of data to a HDF5 sink*

Description

Write chunks of data to a HDF5 sink

Usage

```
.write_chunks_to_hdf5(temp_chunk_dirs, files_in_chunks, hdf5_sink, hdf5_grid)
```

Arguments

temp_chunk_dirs	A vector giving the temporary directory associated with each chunk.
files_in_chunks	A list of files associated with each chunk in the order they should be placed.
hdf5_sink	A HDF5RealizationSink.
hdf5_grid	A RegularArrayGrid.

Value

Invisibly returns TRUE.

`annotateGRanges` *Annotate GRanges*

Description

Annotate GRanges

Usage

```
annotateGRanges(
  genomic_regions,
  annotation_ranges,
  ignore.strand = TRUE,
  overlap_measure = "absolute"
)
```

Arguments

`genomic_regions`	A GRanges object to be annotated
`annotation_ranges`	A GRangesList object with GRanges for different features e.g. introns, exons, enhancers.
`ignore.strand`	TRUE or FALSE indicating whether strand should be ignored when calculating intersections. Default is TRUE.
`overlap_measure`	One of "absolute", "proportion" or "jaccard" indicating whether to calculate the absolute size of the intersection in base pairs, the proportion of base pairs of genomic_ranges overlapping one of the component GRanges of annotation_ranges. or the Jaccard index of the intersection in terms of base pairs. Default value is "absolute".

Value

A numeric vector with the overlap measure for genomic_regions with each type of region in annotation_ranges.

Examples

```
# Load annotation for CpG islands and repetitive DNA
cpg_island_annotation <- annotatr::build_annotations(genome = "hg38", annotations = "hg38_cpgs")
cpg_island_annotation <- cpg_island_annotation[cpg_island_annotation$type == "hg38_cpg_islands"]
repeat_annotation_hg38 <- AnnotationHub::AnnotationHub()[["AH99003"]]

# Convert repeat_annotation_hg38 into a GRangesList
repeat_annotation_hg38 <- GRangesList(split(repeat_annotation_hg38, repeat_annotation_hg38$repClass))

# Calculate the proportion of base pairs in CpG islands overlapping different classes of repetitive elements
annotateGRanges(genomic_regions = cpg_island_annotation, annotation_ranges = repeat_annotation_hg38, overlap_
```

annotatePlot *Create a plot with genomic annotation for a plot of values at methylation sites.*

Description

Works with plots returned by `plotMethylationValues()`, `plotMethSiteCorCoefs()` or `plotMethodicalScores`. Can combine the meth site values plot and genomic annotation together into a single plot or return the annotation plot separately.

Usage

```
annotatePlot(
  meth_site_plot,
  annotation_grl,
  reference_tss = FALSE,
  grl_colours = NULL,
  annotation_line_size = 5,
  annotation_plot_proportion = 0.5,
  keep_meth_site_plot_legend = FALSE,
  annotation_plot_only = FALSE
)
```

Arguments

`meth_site_plot` A plot of methylation site values (generally methylation level or correlation of methylation with transcription) around a TSS

`annotation_grl` A GRangesList object (or list coercible to a GRangesList) where each component GRanges gives the locations of different classes of regions to display. Each class of region will be given a separate colour in the plot, with regions ordered by the order of `names(annotation_grl)`.

`reference_tss` TRUE or FALSE indicating whether to show distances on the X-axis relative to the TSS stored as an attribute `tss_range` of meth_site_plot. Alternatively, can provide a GRanges object with a single range for such a TSS site. In either case, will show the distance of methylation sites to the start of this region with methylation sites upstream relative to the reference_tss shown first. If FALSE (the default), the x-axis will instead show the start site coordinate of the methylation site. relative to the reference_tss shown first. If not, the x-axis will show the start site coordinate of the methylation site.

`grl_colours` An optional vector of colours used to display each of the GRanges making up annotation_grl. Must have same length as annotation_grl.

`annotation_line_size`
Linewidth for annotation plot. Default is 5.

`annotation_plot_proportion`
A value giving the proportion of the height of the plot devoted to the annotation. Default is 0.5.

`keep_meth_site_plot_legend`
TRUE or FALSE indicating whether to retain the legend of meth_site_plot, if it has one. Default value is FALSE.

annotation_plot_only
: TRUE or FALSE indicating whether to return only the annotation plot. Default is to combine meth_site_plot with the annotation.

Value

A ggplot object

Examples

```
# Get CpG islands from UCSC
cpg_island_annotation <- annotatr::build_annotations(genome = "hg38", annotations = "hg38_cpgs")
cpg_island_annotation <- GRangesList(split(cpg_island_annotation, cpg_island_annotation$type))

# Load plot with CpG methylation correlation values for TUBB6
data("tubb6_correlation_plot", package = "methodical")

# Add positions of CpG islands to tubb6_correlation_plot
methodical::annotatePlot(tubb6_correlation_plot, annotation_grl = cpg_island_annotation, annotation_plot_pro
```

---

calculateMethSiteTranscriptCors

Calculate correlation between expression of transcripts and methylation of sites surrounding their TSS

Description

Calculate correlation between expression of transcripts and methylation of sites surrounding their TSS

Usage

```
calculateMethSiteTranscriptCors(
  meth_rse,
  assay_number = 1,
  transcript_expression_table,
  samples_subset = NULL,
  tss_gr,
  expand_upstream = 5000,
  expand_downstream = 5000,
  cor_method = "pearson",
  add_distance_to_region = TRUE,
  max_sites_per_chunk = NULL,
  BPPARAM = BiocParallel::bpparam(),
  results_dir = NULL
)
```

Arguments

meth_rse	A RangedSummarizedExperiment for methylation sites.
assay_number	The assay from meth_rse to extract values from. Default is the first assay.
transcript_expression_table	A matrix or data.frame with the expression values for transcripts, where row names are transcript names and columns sample names. There should be a row corresponding to each transcript associated with each range in tss_gr. Names of samples must match those in meth_rse unless samples_subset provided.
samples_subset	Sample names used to subset meth_rse and transcript_expression_table. Provided samples must be found in both meth_rse and transcript_expression_table. Default is to use all samples in meth_rse and transcript_expression_table.
tss_gr	A GRanges object with the locations of transcription start sites (TSS). Each region should have a width of 1. names(tss_gr) should give the name of the transcript associated with the TSS, which must be present in transcript_expression_table.
expand_upstream	Number of bases to add upstream of each TSS. Must be numeric vector of length 1 or equal to the length of tss_gr. Default is 5000.
expand_downstream	Number of bases to add downstream of each TSS. Must be numeric vector of length 1 or equal to the length of tss_gr. Default is 5000.
cor_method	A character string indicating which correlation coefficient is to be computed. One of either "pearson" or "spearman" or their abbreviations.
add_distance_to_region	TRUE or FALSE indicating whether to add the distance of methylation sites to the TSS. Default value is TRUE. Setting to FALSE will roughly half the total running time.
max_sites_per_chunk	The approximate maximum number of methylation sites to try to load into memory at once. The actual number loaded may vary depending on the number of methylation sites overlapping each region, but so long as the size of any individual regions is not enormous (>= several MB), it should vary only very slightly. Some experimentation may be needed to choose an optimal value as low values will result in increased running time, while high values will result in a large memory footprint without much improvement in running time. Default is floor(62500000/ncol(meth_rse)), resulting in each chunk requiring approximately 500 MB of RAM.
BPPARAM	A BiocParallelParam object for parallel processing. Defaults to `BiocParallel::bpparam()`.
results_dir	An optional path to a directory to save results as RDS files. There will be one RDS file for each transcript. If not provided, returns the results as a list.

Value

If results_dir is NULL, a list of data.frames with the correlation of methylation sites surrounding a specified genomic region with a given feature, p-values and adjusted q-values for the correlations. Distance of the methylation sites upstream or downstream to the center of the region is also provided. If results_dir is provided, instead returns a list with the paths to the RDS files with the results.

Examples

```
# Load TUBB6 TSS GRanges, RangedSummarizedExperiment with methylation values for CpGs around TUBB6 TSS and TUBB6
data(tubb6_tss, package = "methodical")
data(tubb6_meth_rse, package = "methodical")
tubb6_meth_rse <- eval(tubb6_meth_rse)
data(tubb6_transcript_counts, package = "methodical")

# Calculate correlation values between methylation values and transcript values for TUBB6
tubb6_cpg_meth_transcript_cors <- methodical::calculateMethSiteTranscriptCors(meth_rse = tubb6_meth_rse,
 transcript_expression_table = tubb6_transcript_counts, tss_gr = tubb6_tss, expand_upstream = 5000, expand_do
```

`calculateRegionMethylationTranscriptCors` *Calculate the correlation values between the methylation of genomic regions and the expression of associated transcripts*

Description

Calculate the correlation values between the methylation of genomic regions and the expression of associated transcripts

Usage

```
calculateRegionMethylationTranscriptCors(
  meth_rse,
  assay_number = 1,
  transcript_expression_table,
  samples_subset = NULL,
  genomic_regions,
  genomic_region_names = NULL,
  genomic_region_transcripts = NULL,
  genomic_region_methylation = NULL,
  cor_method = "pearson",
  p_adjust_method = "BH",
  region_methylation_summary_function = colMeans,
  BPPARAM = BiocParallel::bpparam(),
  ...
)
```

Arguments

`meth_rse`	A RangedSummarizedExperiment with methylation values for CpG sites which will be used to calculate methylation values for genomic_regions. There must be at least 3 samples in common between meth_rse and transcript_expression_table.
`assay_number`	The assay from meth_rse to extract values from. Default is the first assay.
`transcript_expression_table`	A table with the expression values for different transcripts in different samples. Row names should give be the transcript name and column names should be the name of samples.

samples_subset Optional sample names used to subset meth_rse and transcript_expression_table. Provided samples must be found in both meth_rse and transcript_expression_table. Default is to use all samples in meth_rse and transcript_expression_table.

genomic_regions
A GRanges object.

genomic_region_names
Names for genomic_regions. If not provided, attempts to use names(genomic_regions).

genomic_region_transcripts
Names of transcripts associated with each region in genomic_regions. If not provided, attempts to use genomic_regions$transcript_id. All transcripts must be present in transcript_expression_table.

genomic_region_methylation
Optional preprovided table with methylation values for genomic_regions such as created using summarizeRegionMethylation(). Table will be created if it is not provided which will increase running time. Row names should match genomic_region_names and column names should match those of transcript_expression_table

cor_method A character string indicating which correlation coefficient is to be computed. One of either "pearson" or "spearman" or their abbreviations.

p_adjust_method
Method used to adjust p-values. Same as the methods from p.adjust.methods. Default is Benjamini-Hochberg.

region_methylation_summary_function
A function that summarizes column values. Default is colMeans.

BPPARAM A BiocParallelParam object for parallel processing. Defaults to `BiocParallel::bpparam()`.

... Additional arguments to be passed to summary_function.

Value

A data.frame with the correlation values between the methylation of genomic regions and expression of transcripts associated with them

Examples

```
# Load TUBB6 TMRs, RangedSummarizedExperiment with methylation values for CpGs around TUBB6 TSS and TUBB6 transc
data(tubb6_tmrs, package = "methodical")
data(tubb6_meth_rse, package = "methodical")
tubb6_meth_rse <- eval(tubb6_meth_rse)
data(tubb6_transcript_counts, package = "methodical")

# Calculate correlation values between TMRs identified for TUBB6 and transcript expression
tubb6_tmrs_transcript_cors <- methodical::calculateRegionMethylationTranscriptCors(
  meth_rse = tubb6_meth_rse, transcript_expression_table = tubb6_transcript_counts,
  genomic_regions = tubb6_tmrs, genomic_region_names = tubb6_tmrs$tmr_name)
tubb6_tmrs_transcript_cors
```

`calculateSmoothedMethodicalScores` *Calculate methodical score and smooth it using a exponential weighted moving average*

Description

Calculate methodical score and smooth it using a exponential weighted moving average

Usage

```
calculateSmoothedMethodicalScores(
  correlation_df,
  offset_length = 10,
  smoothing_factor = 0.75
)
```

Arguments

`correlation_df`	A data.frame with correlation values between methylation sites and a transcript as returned by calculateMethSiteTranscriptCors.
`offset_length`	Number of methylation sites added upstream and downstream of a central methylation site to form a window, resulting in a window size of 2*offset_length + 1. Default value is 10.
`smoothing_factor`	Smoothing factor for exponential moving average. Should be a value between 0 and 1 with higher values resulting in a greater degree of smoothing. Default is 0.75.

Value

A GRanges object

Examples

```
# Load data.frame with CpG methylation-transcription correlation results for TUBB6
data("tubb6_cpg_meth_transcript_cors", package = "methodical")

# Calculate smoothed Methodical scores from correlation values
smoothed_methodical_scores <- methodical::calculateSmoothedMethodicalScores(tubb6_cpg_meth_transcript_cors)
```

`createRandomRegions` *Create a GRanges with random regions from a genome*

Description

Can constrain the random regions so that they do not overlap each other and/or an optional set of masked regions. Random regions which do meet these constraints will be discarded and new ones generated until the desired number of regions has been reached or a maximum allowed number of attempts has been made. After the maximum number of allowed attempts, the created random regions meeting the constraints up to that point will be returned. Any random regions that are out-of-bounds relative to their sequence length are trimmed before being returned.

Usage

```
createRandomRegions(
  genome,
  n_regions = 1000,
  region_widths = 1000,
  sequences = NULL,
  all_sequences_equally_likely = FALSE,
  stranded = FALSE,
  masked_regions = NULL,
  allow_overlapping_regions = FALSE,
  ignore.strand = TRUE,
  max_tries = 100
)
```

Arguments

`genome`	A BSgenome object.
`n_regions`	Number of random regions to create. Default is 1000.
`region_widths`	The widths of the random regions. Widths cannot be negative. Can be just a single value if all regions are to have the same widths. Default is 1000.
`sequences`	The names of sequences to create random regions on. Default is to use all standard sequences (those without "_" in their name)
`all_sequences_equally_likely`	TRUE or FALSE indicating if the probability of creating random regions on a sequence should be the same for each sequence. Default is FALSE, indicating to make the probability proportional to a sequences length.
`stranded`	TRUE or FALSE indicating if created regions should have a strand randomly assigned. Default is FALSE, indicating to make unstranded regions.
`masked_regions`	An optional GRanges object which random regions will not be allowed to overlap.
`allow_overlapping_regions`	TRUE or FALSE indicating if created random regions should be allowed to overlap. Default is FALSE.
`ignore.strand`	TRUE or FALSE indicating whether strand should be ignored when identifying overlaps between random regions with each other or with masked_regions. Only relevant if stranded is TRUE and either allow_overlapping_regions is FALSE or masked_regions is provided. Default is TRUE.

max_tries The maximum number of attempts to make to find non-overlapping regions which do not overlap masked_regions. Default value is 100.

Value

A GRanges object

Examples

```
# Set random seed
set.seed(123)

# Create 10,000 random non-overlapping regions with width 1,000 for hg38
random_regions <- methodical::createRandomRegions(genome = "BSgenome.Hsapiens.UCSC.hg38", n_regions = 10000)
```

---

extractGRangesMethSiteValues

Extract values for methylation sites overlapping genomic regions from a methylation RSE.

Description

Extract values for methylation sites overlapping genomic regions from a methylation RSE.

Usage

```
extractGRangesMethSiteValues(
  meth_rse,
  genomic_regions = NULL,
  samples_subset = NULL,
  assay_number = 1
)
```

Arguments

meth_rse	A RangedSummarizedExperiment for methylation data.
genomic_regions	A GRanges object. If set to NULL, returns all methylation sites in meth_rse
samples_subset	Optional sample names used to subset meth_rse.
assay_number	The assay from meth_rse to extract values from. Default is the first assay.

Value

A data.frame with the methylation site values for all sites in meth_rse which overlap genomic_ranges. Row names are the coordinates of the sites as a character vector.

Examples

```
# Load sample RangedSummarizedExperiment with CpG methylation data
data(tubb6_meth_rse, package = "methodical")
tubb6_meth_rse <- eval(tubb6_meth_rse)

# Create a sample GRanges object to use
test_region <- GRanges("chr18:12305000-12310000")

# Get methylation values for CpG sites overlapping HDAC1 gene
test_region_methylation <- methodical::extractGRangesMethSiteValues(meth_rse = tubb6_meth_rse, genomic_regio
```

`extractMethSitesFromGenome` *Create a GRanges with methylation sites of interest from a BSgenome.*

Description

Create a GRanges with methylation sites of interest from a BSgenome.

Usage

```
extractMethSitesFromGenome(
  genome,
  pattern = "CG",
  plus_strand_only = TRUE,
  meth_site_position = 1,
  standard_sequences_only = TRUE
)
```

Arguments

`genome`	A BSgenome object (or the name of one) or a DNAStringSet with names indicating the sequences.
`pattern`	A pattern to match in bsgenome. Default is "CG".
`plus_strand_only`	TRUE or FALSE indicating whether to only return matches on "+" strand, avoiding returning duplicate hits for palindromic sequences e.g. CG. Not relevant if genome is a DNAStringSet. Default is TRUE.
`meth_site_position`	Which position in the pattern corresponds to the methylation site of interest. Default is the first position.
`standard_sequences_only`	TRUE or FALSE indicating whether to only return sites on standard sequences (those without "-" in their names). Default is TRUE.

Value

A GRanges object with genomic regions matching the pattern.

Examples

```
# Get human CpG sites for hg38 genome build
hg38_cpgs <- methodical::extractMethSitesFromGenome("BSgenome.Hsapiens.UCSC.hg38")

# Find CHG sites in Arabidopsis thaliana
arabidopsis_cphpgs <- methodical::extractMethSitesFromGenome("BSgenome.Athaliana.TAIR.TAIR9", pattern = "CHG
```

`findTMRs` *Find TSS-Proximal Methylation-Controlled Regulatory Sites (TMRs)*

Description

Find TSS-Proximal Methylation-Controlled Regulatory Sites (TMRs)

Usage

```
findTMRs(
  correlation_df,
  offset_length = 10,
  smoothing_factor = 0.75,
  p_value_threshold = 0.005,
  min_gapwidth = 150,
  min_meth_sites = 5
)
```

Arguments

`correlation_df`	A data.frame with correlation values between methylation sites and a transcript or a path to an RDS file containing such a data.frame as returned by calculateMethSiteTranscriptCors.
`offset_length`	Number of methylation sites added upstream and downstream of a central methylation site to form a window, resulting in a window size of 2*offset_length + 1. Default value is 10.
`smoothing_factor`	Smoothing factor for exponential moving average. Should be a value between 0 and 1 with higher values resulting in a greater degree of smoothing. Default is 0.75.
`p_value_threshold`	The p_value cutoff to use. Default value is 0.005
`min_gapwidth`	Merge TMRs with the same direction separated by less than this number of base pairs. Default value is 150.
`min_meth_sites`	Minimum number of methylation sites that TMRs can contain. Default value is 5.

Value

A GRanges object with the location of TMRs.

Examples

```
# Load methylation-transcript correlation results for TUBB6 gene
data("tubb6_cpg_meth_transcript_cors", package = "methodical")

# Find TMRs for
tubb6_tmrs <- findTMRs(correlation_df = tubb6_cpg_meth_transcript_cors)
print(tubb6_tmrs)
```

`hg38_cpgs_subset` *hg38_cpgs_subset*

Description

All the CpG sites within the first one million base pairs of chromosome 1.

Usage

```
hg38_cpgs_subset
```

Format

A GRanges object.

`infinium_450k_probe_granges_hg19`
infinium_450k_probe_granges_hg19

Description

The hg19 genomic coordinates for methylation sites analysed by the Infinium HumanMethylation450K array.

Usage

```
infinium_450k_probe_granges_hg19
```

Format

GRanges object with 482,421 ranges and one metadata column name giving the name of the associated probe.

Source

Derived from the manifest file downloaded from https://webdata.illumina.com/downloads/productfiles/humanmethylation
2.csv?_gl<-1*ocsx4f*_ga*MTk1Nzc4MDkwMy4xNjg3ODcxNjg0*_ga_VVVPY8BDYL*MTY4Nzg3MTY4My4xLjEuMTY

kallistoIndex *Create an index file for running Kallisto*

Description

Create an index file for running Kallisto

Usage

```
kallistoIndex(
  path_to_kallisto,
  transcripts_fasta,
  index_name = "kallistoIndex.idx"
)
```

Arguments

path_to_kallisto	Path to kallisto executable
transcripts_fasta	Path to a fasta file for the transcripts to be quantified.
index_name	Name to give the created index file. Default is "kallistoIndex.idx".

Value

Invisibly returns TRUE.

Examples

```
## Not run:
# Download transcripts FASTA from Gencode
download.file("https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_44/gencode.v44.transcripts

# Locate the kallisto executable (provided that it is on the path)
kallisto_path <- system2("which", args = "kallisto", stdout = TRUE)

# Create transcripts index for use with Kallisto
methodical::kallistoIndex(kallisto_path, transcripts_fasta = "gencode.v44.transcripts.fa.gz")

## End(Not run)
```

kallistoQuantify *Run kallisto on a set of FASTQ files and merge the results*

Description

Run kallisto on a set of FASTQ files and merge the results

Usage

```
kallistoQuantify(
  path_to_kallisto,
  kallistoIndex,
  forward_fastq_files,
  reverse_fastq_files,
  sample_names,
  output_directory,
  merged_output_prefix = "kallisto_transcript",
  messages_file = "",
  ncores = 1,
  number_bootstraps = 100
)
```

Arguments

path_to_kallisto	Path to kallisto executable
kallistoIndex	Path to a kallisto index
forward_fastq_files	A vector with the paths to forward FASTQ files. Each file should correspond to the file at the same position in reverse_fastq_files.
reverse_fastq_files	A vector with the paths to reverse FASTQ files. Each file should correspond to the file at the same position in forward_fastq_files.
sample_names	A vector with the names of samples for each pair of samples from forward_fastq_files and reverse_fastq_files
output_directory	The name of the directory to save results in. Will be created if it doesn't exist.
merged_output_prefix	Prefix to use for names of merged output files for counts and TPM which take the form merged_output_prefix_counts_merged.tsv.gz and merged_output_prefix_tpm_merged.tsv.gz. Default prefix is "kallisto_transcript" i.e. default output merged output files are kallisto_transcript_counts_merged.tsv.gz and kallisto_transcript_tpm_merged.tsv.gz.
messages_file	Name of file to save kallisto run messages. If no file name given, information is printed to stdout.
ncores	The number of cores to use. Default is 1.
number_bootstraps	The number of bootstrap samples. Default is 100.

Value

The path to the merged counts table.

`liftoverMethRSE` *Liftover rowRanges of a RangedSummarizedExperiment for methylation data from one genome build to another*

Description

Removes methylation sites which cannot be mapped to the target genome build and those which result in many-to-one mappings. Also removes one-to-many mappings by default and can remove sites which do not map to allowed regions in the target genome e.g. CpG sites.

Usage

```
liftoverMethRSE(
  meth_rse,
  chain,
  remove_one_to_many_mapping = TRUE,
  permitted_target_regions = NULL
)
```

Arguments

`meth_rse`	A RangedSummarizedExperiment for methylation data
`chain`	A "Chain" object to be used with rtracklayer::liftOver
`remove_one_to_many_mapping`	TRUE or FALSE indicating whether to remove regions in the source genome which map to multiple regions in the target genome. Default is TRUE.
`permitted_target_regions`	An optional GRanges object used to filter the rowRanges by overlaps after liftover, for example CpG sites from the target genome. Any regions which do not overlap permitted_target_regions will be removed. GRangesList to GRanges if all remaining source regions can be uniquely mapped to the target genome.

Value

A RangedSummarizedExperiment with rowRanges lifted over to the genome build indicated by chain.

Examples

```
# Load sample RangedSummarizedExperiment with CpG methylation data
data(tubb6_meth_rse, package = "methodical")
tubb6_meth_rse <- eval(tubb6_meth_rse)

# Get CpG sites for hg19
hg19_cpgs <- methodical::extractMethSitesFromGenome("BSgenome.Hsapiens.UCSC.hg19")

# Get liftover chain for mapping hg38 to hg19
library(AnnotationHub)
ah <- AnnotationHub()
chain <- ah[["AH14108"]]

# Liftover tubb6_meth_rse from hg38 to hg19, keeping only sites that were mapped to CpG sites in hg19
```

```
tubb6_meth_rse_hg19 <- methodical::liftoverMethRSE(tubb6_meth_rse, chain = chain,
  permitted_target_regions = hg19_cpgs)
```

`makeMethRSEFromArrayFiles` *Create a HDF5-backed RangedSummarizedExperiment for methylation values in array files*

Description

Create a HDF5-backed RangedSummarizedExperiment for methylation values in array files

Usage

```
makeMethRSEFromArrayFiles(
  array_files,
  probe_name_column = 1,
  beta_value_column = 2,
  normalization_factor = NULL,
  decimal_places = NA,
  probe_ranges,
  sample_metadata = NULL,
  hdf5_dir,
  dataset_name = "beta",
  overwrite = FALSE,
  chunkdim = NULL,
  temporary_dir = NULL,
  BPPARAM = BiocParallel::bpparam(),
  ...
)
```

Arguments

`array_files`	A vector of paths to bedGraph files. Automatically detects if array_files contain a header if every field in the first line is a character.
`probe_name_column`	The number of the column which corresponds to the name of the probes. Default is 1st column.
`beta_value_column`	The number of the column which corresponds to the beta values . Default is 2nd column.
`normalization_factor`	An optional numerical value to divide methylation values by to convert them to fractions e.g. 100 if they are percentages. Default is not to leave values as they are in the input files.
`decimal_places`	Integer indicating the number of decimal places to round beta values to. Default is 2.
`probe_ranges`	A GRanges object giving the genomic locations of probes where each region corresponds to a separate probe. There should be a metadata column called name with the name of the probe associated with each region. Any probes in array_files that are not in probe_ranges are ignored.

sample_metadata	Sample metadata to be used as colData for the RangedSummarizedExperiment
hdf5_dir	Directory to save HDF5 file. Is created if it doesn't exist. HDF5 file is called assays.h5.
dataset_name	Name to give data set in HDF5 file. Default is "beta".
overwrite	TRUE or FALSE indicating whether to allow overwriting if dataset_name already exists in assays.h5. Default is FALSE.
chunkdim	The dimensions of the chunks for the HDF5 file. Should be a vector of length 2 giving the number of rows and then the number of columns in each chunk.
temporary_dir	Name to give a temporary directory to store intermediate files. A directory with this name cannot already exist. Default is to create a name using tempfile("temporary_meth_chunks_").
BPPARAM	A BiocParallelParam object for parallel processing. Defaults to `BiocParallel::bpparam()`.
...	Additional arguments to be passed to HDF5Array::HDF5RealizationSink() for controlling the physical properties of the created HDF5 file, such as compression level. Uses the defaults for any properties that are not specified.

Value

A RangedSummarizedExperiment with methylation values for all methylation sites in meth_sites. Methylation sites will be in the same order as sort(meth_sites).

Examples

```
# Get human CpG sites for hg38 genome build
data("infinium_450k_probe_granges_hg19", package = "methodical")

# Get paths to array files
array_files <- list.files(path = system.file('extdata', package = 'methodical'),
  pattern = ".txt.gz", full.names = TRUE)

# Create sample metadata
sample_metadata <- data.frame(
  tcga_project = "LUAD",
  sample_type = "Tumour", submitter = gsub("_01.tsv.gz", "", basename(array_files)),
  row.names = gsub(".tsv.gz", "", basename(array_files))
)

# Create a HDF5-backed RangedSummarizedExperiment from array files using default chumk dimensions
meth_rse <- makeMethRSEFromArrayFiles(array_files = array_files,
 probe_ranges = infinium_450k_probe_granges_hg19,
 sample_metadata = sample_metadata, hdf5_dir =  paste0(tempdir(), "/array_file_hdf5_1"))
```

---

makeMethRSEFromBedgraphs

Create a HDF5-backed RangedSummarizedExperiment for methylation values in bedGraphs

---

Description

Create a HDF5-backed RangedSummarizedExperiment for methylation values in bedGraphs

Usage

```
makeMethRSEFromBedgraphs(
  bedgraphs,
  seqnames_column = 1,
  start_column = 2,
  end_column = 3,
  value_column = 4,
  zero_based = TRUE,
  normalization_factor = NULL,
  decimal_places = NA,
  meth_sites,
  sample_metadata = NULL,
  hdf5_dir,
  dataset_name = "beta",
  overwrite = FALSE,
  chunkdim = NULL,
  temporary_dir = NULL,
  BPPARAM = BiocParallel::bpparam(),
  ...
)
```

Arguments

bedgraphs	A vector of paths to bedGraph files. Automatically detects if bedGraphs contain a header if every field in the first line is a character.
seqnames_column	The column number in bedgraphs which corresponds to the sequence names. Default is 1st column.
start_column	The column number in bedgraphs which corresponds to the start positions. Default is 2nd column.
end_column	The column number in bedgraphs which corresponds to the end positions. Default is 3rd column.
value_column	The column number in bedgraphs which corresponds to the methylation values. Default is 4th column.
zero_based	TRUE or FALSE indicating if files are zero-based. Default value is TRUE.
normalization_factor	An optional numerical value to divide methylation values by to convert them to fractions e.g. 100 if they are percentages. Default is not to leave values as they are in the input files.
decimal_places	Optional integer indicating the number of decimal places to round beta values to. Default is not to round.
meth_sites	A GRanges object with the locations of the methylation sites of interest. Any methylation sites in bedGraphs that are not in meth_sites are ignored.
sample_metadata	Sample metadata to be used as colData for the RangedSummarizedExperiment.
hdf5_dir	Directory to save HDF5 file. Is created if it doesn't exist. HDF5 file is called assays.h5.
dataset_name	Name to give data set in HDF5 file. Default is "beta".

overwrite	TRUE or FALSE indicating whether to allow overwriting if dataset_name already exists in assays.h5. Default is FALSE.
chunkdim	The dimensions of the chunks for the HDF5 file. Should be a vector of length 2 giving the number of rows and then the number of columns in each chunk. Uses HDF5Array::getHDF5DumpChunkDim(length(meth_sites), length(bedgraphs))) by default.
temporary_dir	Name to give temporary directory created to store intermediate files. A directory with this name cannot already exist. Default is to create a name using tempfile("temporary_meth_chunks_"). Will be deleted after completion.
BPPARAM	A BiocParallelParam object for parallel processing. Defaults to `BiocParallel::bpparam()`.
...	Additional arguments to be passed to HDF5Array::HDF5RealizationSink() for controlling the physical properties of the created HDF5 file, such as compression level. Uses the defaults for any properties that are not specified.

Value

A RangedSummarizedExperiment with methylation values for all methylation sites in meth_sites. methylation sites will be in the same order as sort(meth_sites).

Examples

```
# Load CpGs within first million base pairs of chromosome 1 as a GRanges object
data("hg38_cpgs_subset", package = "methodical")

# Get paths to bedGraphs
bedgraphs <- list.files(path = system.file('extdata', package = 'methodical'),
  pattern = ".bg.gz", full.names = TRUE)

# Create sample metadata
sample_metadata <- data.frame(
  tcga_project = gsub("_.*", "", gsub("TCGA_", "", basename(bedgraphs))),
  sample_type = ifelse(grepl("N", basename(bedgraphs)), "Normal", "Tumour"),
  row.names = tools::file_path_sans_ext(basename(bedgraphs))
)

# Create a HDF5-backed RangedSummarizedExperiment from bedGraphs
meth_rse <- makeMethRSEFromBedgraphs(bedgraphs = bedgraphs,
  meth_sites = hg38_cpgs_subset, sample_metadata = sample_metadata,
  hdf5_dir = paste0(tempdir(), "/bedgraph_hdf5_1"))
```

---

maskRangesInRSE *Mask regions in a ranged summarized experiment*

Description

Mask regions in a ranged summarized experiment

Usage

```
maskRangesInRSE(rse, mask_ranges, assay_number = 1)
```

Arguments

rse	A RangedSummarizedExperiment.
mask_ranges	Either a GRanges with regions to be masked in all samples (e.g. repetitive sequences) or a GRangesList object with different regions to mask in each sample (e.g. mutations). If using a GRangesList object, names of the list elements should be the names of samples in rse.
assay_number	Assay to perform masking. Default is first assay

Value

A RangedSummarizedExperiment with the regions present in mask_ranges masked

Examples

```
# Load sample RangedSummarizedExperiment with CpG methylation data
data(tubb6_meth_rse, package = "methodical")
tubb6_meth_rse <- eval(tubb6_meth_rse)

# Create a sample GRanges object to use to mask tubb6_meth_rse
mask_ranges <- GRanges("chr18:12305000-12310000")

# Mask regions in tubb6_meth_rse
tubb6_meth_rse_masked <- methodical::maskRangesInRSE(tubb6_meth_rse, mask_ranges)

# Count the number of NA values before and after masking
sum(is.na(assay(tubb6_meth_rse)))
sum(is.na(assay(tubb6_meth_rse_masked)))
```

---

methrixToRSE *Convert a Methrix object into a RangedSummarizedExperiment*

Description

Convert a Methrix object into a RangedSummarizedExperiment

Usage

```
methrixToRSE(methrix, assays = c("beta", "cov"))
```

Arguments

methrix	A methrix object
assays	A vector indicating the names of assays in methrix used to create a RangedSummarizedExperiment. Can be one or both of "beta" and "cov". Default is both "beta" and "cov" assays.

Value

A RangedSummarizedExperiment

Examples

```
# Load a sample methrix object
data("methrix_data", package = "methrix")

# Convert methrix to a RangedSummarizedExperiment with one assay for the methylation beta values
meth_rse <- methodical::methrixToRSE(methrix_data, assays = "beta")
```

`plotMethodicalScores` *Create plot of Methodical score values for methylation sites around a TSS*

Description

Create plot of Methodical score values for methylation sites around a TSS

Usage

```
plotMethodicalScores(
  meth_site_values,
  reference_tss = NULL,
  p_value_threshold = 0.005,
  smooth_scores = TRUE,
  offset_length = 10,
  smoothing_factor = 0.75,
  smoothed_curve_colour = "black",
  linewidth = 1,
  curve_alpha = 0.75,
  title = NULL,
  xlabel = "Genomic Position",
  low_colour = "#7B5C90",
  high_colour = "#BFAB25"
)
```

Arguments

`meth_site_values`
: A data.frame with correlation values for methylation sites. There should be one column called "cor". and another called "p_val" which are used to calculate the Methodical score. row.names should be the names of methylation sites and all methylation sites must be located on the same sequence.

`reference_tss`
: An optional GRanges object with a single range. If provided, the x-axis will show the distance of methylation sites to the start of this region with methylation sites upstream. relative to the reference_tss shown first. If not, the x-axis will show the start site coordinate of the methylation site.

`p_value_threshold`
: The p-value threshold used to identify TMRs. Default value is 0.005. Set to NULL to turn off significance thresholds.

`smooth_scores`
: TRUE or FALSE indicating whether to display a curve of smoothed Methodical scores on top of the plot. Default is TRUE.

offset_length	Offset length to be supplied to calculateSmoothedMethodicalScores.
smoothing_factor	Smoothing factor to be provided to calculateSmoothedMethodicalScores.
smoothed_curve_colour	Colour of the smoothed curve. Default is "black".
linewidth	Line width of the smoothed curve. Default value is 1.
curve_alpha	Alpha value for the curve. Default value is 0.75.
title	Title of the plot. Default is no title.
xlabel	Label for the X axis in the plot. Default is "Genomic Position".
low_colour	Colour to use for low values. Default value is "#7B5C90".
high_colour	Colour to use for high values. Default value is "#BFAB25".

Value

A ggplot object

Examples

```
# Load methylation-transcript correlation results for TUBB6 gene
data("tubb6_cpg_meth_transcript_cors", package = "methodical")

# Calculate and plot Methodical scores from correlation values
methodical::plotMethodicalScores(tubb6_cpg_meth_transcript_cors, reference_tss = attributes(tubb6_cpg_meth_t
```

---

plotMethSiteCorCoefs *Plot the correlation coefficients for methylation sites within a region and an associated feature of interest*

Description

Plot the correlation coefficients for methylation sites within a region and an associated feature of interest

Usage

```
plotMethSiteCorCoefs(
  meth_site_cor_values,
  reference_tss = FALSE,
  title = NULL,
  xlabel = NULL,
  ylabel = "Correlation Coefficient",
  value_colours = "set2",
  reverse_x_axis = FALSE
)
```

Arguments

meth_site_cor_values	A data.frame with correlation values associated with methylation sites, such as returned by calculateMethSiteTranscriptCors. There should be one column called meth_site giving the coordinates of methylation sites in character format and another column called cor giving the correlation between the methylation values of the methylation sites and a feature of interest. All methylation sites must be located on the same sequence.
reference_tss	TRUE or FALSE indicating whether to show distances on the X-axis relative to the TSS stored as an attribute tss_range of meth_site_cor_values. Alternatively, can provide a GRanges object with a single range for such a TSS site. In either case, will show the distance of methylation sites to the start of this region with methylation sites upstream relative to the reference_tss shown first. If FALSE (the default), the x-axis will instead show the start site coordinate of the methylation site.
title	Title of the plot. Default is no title.
xlabel	Label for the X axis in the plot. Defaults to "Distance to TSS" if reference_tss is used or "seqname position" where seqname is the name of the relevant sequence.
ylabel	Label for the Y axis in the plot. Default is "Correlation Coefficient".
value_colours	A vector with two colours to use, one for low values and the other for high values. Alternatively, can use one of two predefined colour sets by providing either "set1" or "set2": set1 uses "#53868B" (blue) for low values and "#CD2626" (red) for high values while set2 uses "#7B5C90" (purple) for low values and ""#bfab25" (gold) for high values. Default is "set2".
reverse_x_axis	TRUE or FALSE indicating whether x-axis should be reversed, for example if plotting a region on the reverse strand so that left side of plot corresponds to upstream.

Value

A ggplot object

Examples

```
# Load methylation-transcript correlation results for TUBB6 gene
data("tubb6_cpg_meth_transcript_cors", package = "methodical")

# Plot methylation-transcript correlation values around TUBB6 TSS
methodical::plotMethSiteCorCoefs(tubb6_cpg_meth_transcript_cors, ylabel = "Spearman Correlation")

# Create same plot but showing the distance to the TUBB6 TSS on the x-axis
methodical::plotMethSiteCorCoefs(tubb6_cpg_meth_transcript_cors,
 ylabel = "Spearman Correlation", reference_tss = attributes(tubb6_cpg_meth_transcript_cors)$tss_range)
```

`plotMethylationValues` *Create a plot of methylation values for methylation sites in a region*

Description

Create a plot of methylation values for methylation sites in a region

Usage

```
plotMethylationValues(
  meth_site_values,
  sample_name = NULL,
  reference_tss = FALSE,
  title = NULL,
  xlabel = NULL,
  ylabel = "Methylation Value",
  value_colours = "set1",
  reverse_x_axis = FALSE
)
```

Arguments

`meth_site_values`	A data.frame with values associated with methylation sites. Row names should be the coordinates of methylation sites in character format. All methylation sites must be located on the same sequence.
`sample_name`	Name of column in meth_site_values to plot. Defaults to first column if none provided.
`reference_tss`	TRUE or FALSE indicating whether to show distances on the X-axis relative to the TSS stored as an attribute `tss_range` of meth_site_values. Alternatively, can provide a GRanges object with a single range for such a TSS site. In either case, will show the distance of methylation sites to the start of this region with methylation sites upstream relative to the reference_tss shown first. If FALSE (the default), the x-axis will instead show the start site coordinate of the methylation site.
`title`	Title of the plot. Default is no title.
`xlabel`	Label for the X axis in the plot. Defaults to "Distance to TSS" if reference_tss is used or "seqname position" where seqname is the name of the relevant sequence.
`ylabel`	Label for the Y axis in the plot. Default is "Methylation Value".
`value_colours`	A vector with two colours to use, one for low values and the other for high values. Alternatively, can use one of two predefined colour sets by providing either "set1" or "set2": set1 uses "#53868B" (blue) for low values and "#CD2626" (red) for high values while set2 uses "#7B5C90" (purple) for low values and ""#bfab25" (gold) for high values. Default is "set1".
`reverse_x_axis`	TRUE or FALSE indicating whether x-axis should be reversed, for example if plotting a region on the reverse strand so that left side of plot corresponds to upstream.

Value

A ggplot object

Examples

```
# Load methylation-values around the TUBB6 TSS
data("tubb6_meth_rse", package = "methodical")
tubb6_meth_rse <- eval(tubb6_meth_rse)

# Extract methylation values from tubb6_meth_rse
tubb6_methylation_values = methodical::extractGRangesMethSiteValues(meth_rse = tubb6_meth_rse)

# Plot methylation values around TUBB6 TSS
methodical::plotMethylationValues(tubb6_methylation_values, sample_name = "N1")

# Create same plot but showing the distance to the TUBB6 TSS on the x-axis
data("tubb6_tss", package = "methodical")
methodical::plotMethylationValues(tubb6_methylation_values, sample_name = "N1",
  reference_tss = tubb6_tss)
```

---

plotTMRs *Add TMRs to a methylation site value plot*

Description

Add TMRs to a methylation site value plot

Usage

```
plotTMRs(
  meth_site_plot,
  tmrs_gr,
  reference_tss = NULL,
  transcript_id = NULL,
  tmr_colours = c("#A28CB1", "#D2C465"),
  linewidth = 5
)
```

Arguments

meth_site_plot	A plot of Value around a TSS.
tmrs_gr	A GRanges object giving the position of TMRs.
reference_tss	An optional GRanges object with a single range. If provided, the x-axis will show the distance of methylation sites to the start of this region with methylation sites upstream relative to the reference_tss shown first. If not, the x-axis will show the start site coordinate of the methylation site.
transcript_id	An optional transcript ID. If provided, will attempt to filter tmrs_gr and reference_tss using a metadata column called transcript_id with a value identical to the provided one.
tmr_colours	A vector with colours to use for negative and positive TMRs. Defaults to "#7B5C90" for negative and "#BFAB25" for positive TMRs.
linewidth	A numeric value to be provided as linewidth for geom_segment().

Value

A ggplot object

Examples

```
# Load methylation-transcript correlation results for TUBB6 gene
data("tubb6_cpg_meth_transcript_cors", package = "methodical")

# Plot methylation-transcript correlation values around TUBB6 TSS
tubb6_correlation_plot <- methodical::plotMethSiteCorCoefs(tubb6_cpg_meth_transcript_cors, ylabel = "Spearma

# Find TMRs for TUBB6
tubb6_tmrs <- findTMRs(correlation_df = tubb6_cpg_meth_transcript_cors)

# Plot TMRs on top of tubb6_correlation_plot
methodical::plotTMRs(tubb6_correlation_plot, tmrs_gr = tubb6_tmrs)
```

---

rangesRelativeToTSS *Find locations of genomic regions relative to transcription start sites.*

Description

Find locations of genomic regions relative to transcription start sites.

Usage

```
rangesRelativeToTSS(genomic_regions, tss_gr)
```

Arguments

genomic_regions	A GRanges object.
tss_gr	A GRanges object with transcription start sites. Each range should have width 1. Upstream and downstream are relative to strand of tss_gr.

Value

A GRanges object where all regions have "relative" as the sequence names and ranges are the location of TMRs relative to the TSS.

Examples

```
# Create query and subject GRanges
genomic_regions <- GenomicRanges::GRanges(c("chr1:100-1000:+", "chr1:2000-3000:-"))
tss_gr <- GenomicRanges::GRanges(c("chr1:1500:+", "chr1:4000:-"))

# Calculate distances between query and subject
methodical::rangesRelativeToTSS(genomic_regions, tss_gr)
```

`rapidCorTest` *Rapidly calculate the correlation and the significance of pairs of columns from two data.frames*

Description

Rapidly calculate the correlation and the significance of pairs of columns from two data.frames

Usage

```
rapidCorTest(
  table1,
  table2,
  cor_method = "pearson",
  table1_name = "table1",
  table2_name = "table2",
  p_adjust_method = "BH",
  n_covariates = 0
)
```

Arguments

`table1`	A data.frame
`table2`	A data.frame
`cor_method`	A character string indicating which correlation coefficient is to be computed. One of either "pearson" or "spearman" or their abbreviations.
`table1_name`	Name to give the column giving the name of features in table1. Default is "table1".
`table2_name`	Name to give the column giving the name of features in table2. Default is "table2".
`p_adjust_method`	Method used to adjust p-values. Same as the methods from p.adjust.methods. Default is Benjamini-Hochberg. Setting to "none" will result in no adjusted p-values being calculated.
`n_covariates`	Number of covariates if calculating partial correlations. Defaults to 0.

Value

A data.frame with the correlation and its significance for all pairs consisting of a variable from table1 and a variable from table2.

Examples

```
# Divide mtcars into two tables
table1 <- mtcars[, 1:5]
table2 <- mtcars[, 6:11]

# Calculate correlation between table1 and table2
cor_results <- methodical::rapidCorTest(table1, table2, cor_method = "spearman",
  table1_name = "feature1", table2_name = "feature2")
head(cor_results)
```

`sampleMethSites` *Randomly sample methylation sites from a methylation RSE.*

Description

Randomly sample methylation sites from a methylation RSE.

Usage

```
sampleMethSites(
  meth_rse,
  n_sites = 1000,
  genomic_ranges_filter = NULL,
  invert_filter = FALSE,
  samples_subset = NULL,
  assay_number = 1
)
```

Arguments

`meth_rse`	A RangedSummarizedExperiment for methylation data.
`n_sites`	Number of sites to randomly sample. Default is 1000.
`genomic_ranges_filter`	An optional GRanges object used to first subset meth_rse. Sites will then be chosen randomly from those overlapping these ranges.
`invert_filter`	TRUE or FALSE indicating whether to invert the genomic_ranges_filter so as to exclude sites overlapping these regions. Default value is FALSE.
`samples_subset`	Optional sample names used to subset meth_rse.
`assay_number`	The assay from meth_rse to extract values from. Default is the first assay.

Value

A data.frame with the methylation site values for all sites in meth_rse which overlap genomic_ranges. Row names are the coordinates of the sites as a character vector.

Examples

```
# Load sample RangedSummarizedExperiment with CpG methylation data
data(tubb6_meth_rse, package = "methodical")
tubb6_meth_rse <- eval(tubb6_meth_rse)

# Create a sample GRanges object to use to mask tubb6_meth_rse
mask_ranges <- GRanges("chr18:12305000-12310000")

# Get 20 random CpG sites outside mask_ranges
random_cpgs <- methodical::sampleMethSites(tubb6_meth_rse, n_sites = 20, genomic_ranges_filter = mask_ranges,
  invert_filter = TRUE)

# Check that no CpGs overlap repeats
intersect(rowRanges(random_cpgs), mask_ranges)
```

strandedDistance *Calculate distances of query GRanges upstream or downstream of subject GRanges*

Description

Upstream and downstream are relative to the strand of subject_gr. Unstranded regions are treated the same as regions on the "+" strand.

Usage

```
strandedDistance(query_gr, subject_gr)
```

Arguments

query_gr	A GRanges object
subject_gr	A GRanges object.

Value

A numeric vector of distances

Examples

```
# Create query and subject GRanges
query_gr <- GenomicRanges::GRanges(c("chr1:100-1000:+", "chr1:2000-3000:-"))
subject_gr <- GenomicRanges::GRanges(c("chr1:1500-1600:+", "chr1:4000-4500:-"))

# Calculate distances between query and subject
methodical::strandedDistance(query_gr, subject_gr)
```

summarizeRegionMethylation *Summarize methylation of genomic regions*

Description

Summarize methylation of genomic regions

Usage

```
summarizeRegionMethylation(
  meth_rse,
  assay_number = 1,
  genomic_regions,
  genomic_region_names = NULL,
  keep_metadata_cols = FALSE,
  max_sites_per_chunk = NULL,
  summary_function = base::colMeans,
  na.rm = TRUE,
```

```
  BPPARAM = BiocParallel::bpparam(),
  ...
)
```

Arguments

meth_rse	A RangedSummarizedExperiment with methylation values.
assay_number	The assay from meth_rse to extract values from. Default is the first assay.
genomic_regions	GRanges object with regions to summarize methylation values for.
genomic_region_names	A vector of names to give genomic_regions in the output table. There cannot be any duplicated names. Default is to attempt to use `names(genomic_regions)` if they are present or to name them region_1, region_2, etc otherwise.
keep_metadata_cols	TRUE or FALSE indicating whether to add the metadata columns of genomic_regions to the output. Default is FALSE.
max_sites_per_chunk	The approximate maximum number of methylation sites to try to load into memory at once. The actual number loaded may vary depending on the number of methylation sites overlapping each region, but so long as the size of any individual regions is not enormous (>= several MB), it should vary only very slightly. Some experimentation may be needed to choose an optimal value as low values will result in increased running time, while high values will result in a large memory footprint without much improvement in running time. Default is floor(62500000/ncol(meth_rse)), resulting in each chunk requiring approximately 500 MB of RAM.
summary_function	A function that summarizes column values. Default is base::colMeans.
na.rm	TRUE or FALSE indicating whether to remove NA values when calculating summaries. Default value is TRUE.
BPPARAM	A BiocParallelParam object. Defaults to `BiocParallel::bpparam()`.
...	Additional arguments to be passed to summary_function.

Value

A data.table with the summary of methylation of each region in genomic_regions for each sample.

Examples

```
# Load sample RangedSummarizedExperiment with CpG methylation data
data(tubb6_meth_rse, package = "methodical")
tubb6_meth_rse <- eval(tubb6_meth_rse)

# Create a sample GRanges
test_gr <- GRanges(c("chr18:12303400-12303500", "chr18:12303600-12303750", "chr18:12304000-12306000"))
names(test_gr) <- paste("region", 1:3, sep = "_")

# Calculate mean methylation values for regions in test_gr
test_gr_methylation <- methodical::summarizeRegionMethylation(tubb6_meth_rse, genomic_regions = test_gr,
  genomic_region_names = names(test_gr))
```

sumTranscriptValuesForGenes

Combine the expression values of transcripts to get overall expression of their associated genes

Description

Combine the expression values of transcripts to get overall expression of their associated genes

Usage

```
sumTranscriptValuesForGenes(
  transcript_expression_table,
  gene_to_transcript_list
)
```

Arguments

transcript_expression_table	A table where rows are transcripts and columns are samples. Row names should be the names of transcripts.
gene_to_transcript_list	A list of vectors where the name of each list entry is a gene name and its elements are the names of transcripts. Can alternatively be a GRangeList where the name of each list element is a gene and the names of the individual ranges are transcripts.

Value

A data.frame with the sum of transcript expression values for genes where rows are genes and columns are samples

tubb6_correlation_plot

tubb6_correlation_plot

Description

A plot of the correlation values between methylation-transcription correlations for CpG sites around the TUBB6 TSS.

Usage

```
tubb6_correlation_plot
```

Format

A ggplot object.

tubb6_cpg_meth_transcript_cors *tubb6_cpg_meth_transcript_cors*

Description

A data.frame with the methylation-transcription correlation results for CpGs around the TUBB6 TSS.

A data.frame with the correlation results for CpG sites within +/- 5 KB of the TUBB6 (ENST00000591909) TSS.

Usage

```
tubb6_cpg_meth_transcript_cors

tubb6_cpg_meth_transcript_cors
```

Format

A ggplot object.

A data.frame with 5 columns giving the name of the CpG site (meth_site), name of the transcript associated with the TSS, Spearman correlation value between the methylation of the CpG site and expression of the transcript, p-value associated with the correlations and distance from the CpG site to the TSS.

tubb6_meth_rse *tubb6_meth_rse*

Description

The location of the TSS for TUBB6.

Usage

```
tubb6_meth_rse
```

Format

A call to create a RangedSummarizedExperiment with methylation data for 355 CpG sites within +/- 5,000 base pairs of the TUBB6 TSS in 126 normal prostate samples. Should be evaluated after loading using `tubb6_meth_rse <- tubb6_meth_rse <- eval(tubb6_meth_rse)` to restore the RangedSummarizedExperiment.

Source

WGBS data from 'Li, Jing, et al. "A genomic and epigenomic atlas of prostate cancer in Asian populations." Nature 580.7801 (2020): 93-99.'

tubb6_tmrs *tubb6_tmrs*

Description

TMRs identified for TUBB6

Usage

```
tubb6_tmrs
```

Format

A GRanges object with two ranges.

tubb6_transcript_counts *tubb6_transcript_counts*

Description

Transcript counts for TUBB6 in normal prostate samples.

Usage

```
tubb6_transcript_counts
```

Format

A data.frame with normalized transcript counts for TUBB6 in 126 normal prostate samples.

Source

RNA-seq data from 'Li, Jing, et al. "A genomic and epigenomic atlas of prostate cancer in Asian populations." Nature 580.7801 (2020): 93-99.'

tubb6_tss *tubb6_tss*

Description

The location of the TSS for TUBB6.

Usage

```
tubb6_tss
```

Format

GRanges object with 1 ranges for the TUBB6 TSS.

Source

The TSS of the ENST00000591909 transcript.

TumourMethDatasets *TumourMethDatasets*

Description

A table describing the datasets available from TumourMethData.

Usage

```
TumourMethDatasets
```

Format

A data.frame with one row for each dataset

Index