Package 'decemedip'

May 7, 2026

Title hierarchical Bayesian modeling for cell type deconvolution of immunoprecipitation-based DNA methylome

Version 1.0.0

Description The R package decemedip is a novel computational paradigm developed for inferring the relative abundances of cell types and tissues measure by methylated DNA immunoprecipitation sequencing (MeDIP-Seq). This paradigm allows using reference data from other technologies such as microarray or WGBS.

License MIT + file LICENSE

Encoding UTF-8

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

biocViews Software, ImmunoOncology, DNAMethylation, Epigenetics, Sequencing, WholeGenome

Biarch true

Depends R (>= 4.5.0)

Imports bayesplot, cowplot, dplyr, GenomicRanges, ggplot2, IRanges, magrittr, Matrix, matrixStats, MEDIPS, methods, purrr, R.utils, Rcpp, RcppParallel, rlang, rstan, rstantools, S4Vectors, SummarizedExperiment

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

Suggests knitr, rmarkdown, BiocStyle, devtools, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/nshen7/decemedip

BugReports https://github.com/nshen7/decemedip/issues

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

git_branch RELEASE_3_23

git_last_commit 9e545b6

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-05-07

Author Ning Shen [aut, cre] (ORCID: <https://orcid.org/0000-0002-2974-1086>)

Maintainer Ning Shen <ning.shen.wk@gmail.com>

Contents

`decemedip-package` *The 'decemedip' package.*

Description

The R package decemedip is a novel computational paradigm developed for inferring the relative abundances of cell types and tissues from tissue bulk or circulating cell-free DNA (cfDNA) measure by methylated DNA immunoprecipitation sequencing (MeDIP-Seq). This paradigm allows using reference data from other technologies such as microarray or WGBS.

Author(s)

Maintainer: Ning Shen <ning.shen.wk@gmail.com> (ORCID)

References

Stan Development Team (NA). RStan: the R interface to Stan. R package version 2.32.6. https://mc-stan.org

See Also

Useful links:

- https://github.com/nshen7/decemedip
- Report bugs at https://github.com/nshen7/decemedip/issues

`decemedip` *Main function to perform cell type deconvolution with MeDIP-seq data*

Description

Main function to perform cell type deconvolution with MeDIP-seq data

Usage

```
decemedip(
  sample_bam_file = NULL,
  paired_end = NULL,
  counts_cts = c(),
  counts_anc = c(),
  ref_assembly = "hg19",
  ref_cts = NULL,
  ref_anc = NULL,
  weight_cts = 1,
  weight_anc = 0.5,
  diagnostics = TRUE,
  seed = 2024,
  cores = 1,
  chains = 4,
  iter = 2000,
  stan_input_params = list(s_mu = 3, s_sigma = 3, n_knot_z = 0, degree_z = 3, Xi =
   cor(as.matrix(SummarizedExperiment::assay((ref_cts)))), s_theta = 3, s_tau = 3),
  stan_control = list(adapt_delta = 0.95, max_treedepth = 15),
  timeout_sec = 2 * (diagnostics + 1) * iter * chains,
  max_retries = 3,
  ...
)
```

Arguments

sample_bam_file	A string value that indicates the file path to the bam file of a MeDIP-seq sample of interest. If sample_bam_file is specified, please do not specify counts_cts and counts_anc to avoid conflict.
paired_end	A logic value that indicates whether the bam file is from paired-end reads or single-end. Specify TRUE} for paired-end and \code{FALSE for single-end.
counts_cts	An atomic vector of integer values that indicates the read counts of a MeDIP-seq sample on reference sites/regions. If counts_cts and counts_anc is specified, please do not specify sample_bam_file and paired_end to avoid conflict.
counts_anc	An atomic vector of integer values that indicates the read counts of a MeDIP-seq sample on reference sites/regions. If counts_cts and counts_anc is specified, please do not specify sample_bam_file and paired_end to avoid conflict.
ref_assembly	A string that represents the genome assembly that should be used for cell type-specific sites in the reference panel ('hg19' or 'hg38'). Default to 'hg19'. The default reference is explained in the manuscript of **decemedip**. Alternatively, if the user want to provide their own reference panel by using the ref_cts and ref_anc arguments.
ref_cts	A SummarizedExperiment object that contains the genomic coordinates and beta values of the cell type-specific sites/regions from reference cell types. The \link{makeReferencePanel} can be used to generate such a panel.
ref_anc	Same as ref_cts but for anchor sites.
weight_cts	A numeric value indicating the weights that should be put on cell type-specific sites/regions. Default is 0.5.
weight_anc	A numeric value indicating the weights that should be put on cell type-specific sites/regions. Default is 1.
diagnostics	A logic value that indicates whether to include components of the stan model in the output that are necessary for future diagnostics of the model, such as posterior predictive checks. For details, please refer to the function \link{plotDiagnostics}.
seed	The seed for random number generation in MCMC sampling.
cores	A positive integer specifying the number of cores that can be used for MCMC sampling. The default is 1.
chains	A positive integer specifying the number of Markov chains. The default is 4.
iter	A positive integer specifying the number of iterations for each chain (including warmup). The default is 2000.
stan_input_params	A named list of parameters that specifies the prior of the decemedip model.
stan_control	A named list of parameters to control the sampler's behavior in Stan. See the details in the documentation for the control argument in \link[rstan]{stan}.
timeout_sec	A numerical value indicating the CPU/processor time (in seconds) allowed for the MCMC to run before restarting the chains with a new random seed.
max_retries	An integer value indicating the maximum number of tries with different seed for MCMC if it fails to converge.
...	Other parameters that can be passed to the \link[rstan]{sampling} function.

Value

A list of two elements:

1. data_list: An organized list of variables used as input to the Stan posterior sampling function.
2. posterior: An stanfit object produced by Stan representing the fitted posteriors.

Examples

```
# Here we use a lightweighted example that only contains 8 cell types to avoid long running time:

data(example.hg19.ref.cts.se)
data(example.hg19.ref.anc.se)
data(example.pdx.counts.cts.se)
data(example.pdx.counts.anc.se)

# read counts of cell type-specific CpGs of the sample 'LuCaP_147CR'
counts_cts <- SummarizedExperiment::assays(example.pdx.counts.cts.se)$counts[, "LuCaP_147CR"]
# read counts of anchor CpGs of the sample 'LuCaP_147CR'
counts_anc <- SummarizedExperiment::assays(example.pdx.counts.anc.se)$counts[, "LuCaP_147CR"]

## Fit decemedip model
output <- decemedip(counts_cts = counts_cts, counts_anc = counts_anc,
                    ref_cts = example.hg19.ref.cts.se, ref_anc = example.hg19.ref.anc.se,
                     iter = 500, cores = 1, chains = 1)

## IMPORTANT NOTE (PLEASE READ): For usual cases, you may skip specifying ref_cts and ref_anc,
## as the default panels contain 25 cell types, as used in the manuscript, e.g., run the function as:
## Not run:
  ## By default, the functions uses 2000 iterations, 4 cores and 4 chains
  output <- decemedip(counts_cts = counts_cts, counts_anc = counts_anc)
  ## or use BAM files as input (paired=TRUE is file is paired-end)
  output <- decemedip(sample_bam_file = "path/to/bam/files", paired = TRUE)

## End(Not run)
```

---

example.hg19.ref.anc.se

Subset of hg19.ref.anc.se as a lightweighted example

Description

Subset of hg19.ref.anc.se as a lightweighted example, only contains the blood cell types and prostate.

Usage

```
data(example.hg19.ref.anc.se)
```

Format

An object of class `SummarizedExperiment`.

Details

All coordinates are in hg19.

`example.hg19.ref.cts.se`

Subset of hg19.ref.cts.se as a lightweighted example

Description

Subset of hg19.ref.cts.se as a lightweighted example, only contains the blood cell types and prostate.

Usage

```
data(example.hg19.ref.cts.se)
```

Format

An object of class `SummarizedExperiment`.

Details

All coordinates are in hg19.

`example.pdx.counts.anc.se`

Subset of pdx.counts.anc.se as a lightweighted example

Description

Subset of pdx.counts.anc.se as a lightweighted example, only contains the blood cell types and prostate.

Usage

```
data(example.pdx.counts.anc.se)
```

Format

An object of class `SummarizedExperiment`.

Details

All coordinates are in hg19.

example.pdx.counts.cts.se *Subset of pdx.counts.cts.se as a lightweighted example*

Description

Subset of pdx.counts.cts.se as a lightweighted example, only contains the blood cell types and prostate.

Usage

```
data(example.pdx.counts.cts.se)
```

Format

An object of class `SummarizedExperiment`.

Details

All coordinates are in hg19.

getRoiReadCount *Obtain read counts of regions of interest from MeDIP-seq bam files*

Description

Obtain read counts of regions of interest from MeDIP-seq bam files

Usage

```
getRoiReadCount(
  sample_bam_files,
  sample_names,
  sample_paired,
  roi,
  col_data = NULL,
  row_data = NULL,
  bs_genome = "BSgenome.Hsapiens.UCSC.hg19",
  ...
)
```

Arguments

sample_bam_files	An atomic vector that contains directory paths of the MeDIP-seq sorted bam files.
sample_names	An atomic vector of strings that indicates the sample names. Default is NULL. If not NULL, please make sure that sample_names corresponds to elements of sample_bam_files.
sample_paired	A logic value that indicates sample_paired-end reads (if TRUE) or single-end reads (if FALSE).
roi	A GRanges object that contains the genomic coordinates of the region of interest (ROI).
col_data	A DataFrame object that contains metadata for columns (i.e., samples) if specified. Default is NULL. If not NULL, please make sure that rows of col_data corresponds to elements of sample_bam_files. If input is a non-DataFrame} object, it will be conver
row_data	A DataFrame object that contains metadata for rows (i.e., genomic regions) if specified. Default is NULL. If not NULL, please make sure that rows of row_data corresponds to elements of sample_bam_files. If input is a non-DataFrame} object, it will be converted to a \code{DataFrame.
bs_genome	A character value that indicates the reference genome name as defined by BSgenome package. Default is 'BSgenome.Hsapiens.UCSC.hg19'.
...	Additional arguments passed into MEDIPS::MEDIPS.createROIset

Value

An object of class SummarizedExperiment with read count matrix stored as an assay named 'counts' (can be extracted using SummarizedExperiment::assays)

Examples

```
se <- getRoiReadCount(
  sample_bam_files = c("dir/to/bam1", "dir/to/bam2"),
  sample_names = c("sample1", "sample2"),
  sample_paired = TRUE
)
```

---

getSummaryOnPi *Extract summary statistics and diagnostics on fitted cell type proportions*

Description

Extract summary statistics and diagnostics on fitted cell type proportions

Usage

```
getSummaryOnPi(
  posterior,
  probs = c(0.025, 0.25, 0.5, 0.75, 0.975),
  digits_summary = 5,
  cell_type_names = NULL,
  ...
)
```

Arguments

posterior	The fitted posterior object from decemedip model.
probs	A numeric vector specifying quantiles of interest. The defaults is c(0.025,0.25,0.5,0.75,0.975).
digits_summary	The number of significant digits to use in the summary, defaulting to 5.
cell_type_names	Name of the cell types in reference panel. The order should align with order in the reference panel.
...	Additional arguments that get passed into `rstan::monitor` function.

Value

A data.frame object containg summary statistics and diagnostic statistics of the fitted cell type proportions.

Examples

```
data(pdx.counts.cts.se)
data(pdx.counts.anc.se)
# read counts of cell type-specific CpGs of the sample 'LuCaP_147CR'
counts_cts <- SummarizedExperiment::assays(pdx.counts.cts.se)$counts[, "LuCaP_147CR"]
# read counts of anchor CpGs of the sample 'LuCaP_147CR'
counts_anc <- SummarizedExperiment::assays(pdx.counts.anc.se)$counts[, "LuCaP_147CR"]

## The following functions are commented due to Bioconductor's time constraints on package building
## Fit decemedip model
# output <- decemedip(counts_cts = counts_cts, counts_anc = counts_anc)
## Get summary stats
# smr_pi.df <- getSummaryOnPi(output$posterior)
```

`hg19.ref.anc.se` *Hg19 genomic information of anchor CpGs (i.e., all-tissue unmethylated/methylation probes) inferred from DNA methylation atlas published by Moss 2018 Nat. Commun. (https://www.nature.com/articles/s41467-018-07466-6). Used as default in* `decemedip`.

Description

This dataset represents a GRanges object that contains the collection of Illumina HumanMethylation450K probes that have methylation level less than 0.1 or greater than 0.9 in all tissue present in the atlas. Data source is from the MethAtlas GitHub repo (https://github.com/nloyfer/meth_atlas).

Usage

```
data(hg19.ref.anc.se)
```

Format

An object of class `SummarizedExperiment`.

Details

All coordinates are in hg19.

References

Moss, J., Magenheim, J., Neiman, D. et al. Comprehensive human cell-type methylation atlas reveals origins of circulating cell-free DNA in health and disease. *Nat Commun 9, 5068* (2018).

Examples

```
data(hg19.ref.anc.se)
hg19.ref.anc.se
```

`hg19.ref.cts.se` *Hg19 genomic information of cell-type-specific marker CpGs inferred from DNA methylation atlas published by Moss 2018 Nat. Commun. (https://www.nature.com/articles/s41467-018-07466-6).*

Description

Default reference cell type-specific markers used as default in `decemedip`. This dataset represents a GRanges object that contains the collection of Illumina HumanMethylation450K probes that have methylation level less than 0.1 or greater than 0.9 in all tissue present in the atlas. Data source of the methylation atlas is from the MethAtlas GitHub repo (https://github.com/nloyfer/meth_atlas). For details of how the marker CpGs are selected, please refer to the decemedip manuscript.

Usage

```
data(hg19.ref.cts.se)
```

Format

An object of class `SummarizedExperiment`. rowData(hg19.ref.cts.se) contains information of the selected probes.

Details

All coordinates are in hg19.

References

Moss, J., Magenheim, J., Neiman, D. et al. Comprehensive human cell-type methylation atlas reveals origins of circulating cell-free DNA in health and disease. *Nat Commun 9, 5068* (2018). https://doi.org/10.1038/s41467-018-07466-6

Examples

```
data(hg19.ref.cts.se)
hg19.ref.cts.se
```

`hg38.ref.anc.se` *Hg38 genomic information of cell-type-specific marker CpGs inferred from DNA methylation atlas published by Moss 2018 Nat. Commun. (https://www.nature.com/articles/s41467-018-07466-6).*

Description

Same as `data(hg19.ref.anc.se)` but lifted over to hg38.

Usage

```
data(hg38.ref.anc.se)
```

Format

An object of class `SummarizedExperiment`.

An object of class `SummarizedExperiment`.

Details

All coordinates are in hg38.

Examples

```
data(hg38.ref.anc.se)
hg38.ref.anc.se
```

hg38.ref.cts.se

Hg38 genomic information of cell-type-specific marker CpGs inferred from DNA methylation atlas published by Moss 2018 Nat. Commun. (https://www.nature.com/articles/s41467-018-07466-6).

Description

Same as `data(hg19.ref.cts.se)` but lifted over to hg38.

Usage

```
data(hg38.ref.cts.se)
```

Format

An object of class `SummarizedExperiment`.

An object of class `SummarizedExperiment`.

Details

All coordinates are in hg38.

Examples

```
data(hg38.ref.cts.se)
hg38.ref.cts.se
```

makeReferencePanel

Function for assembling a SummarizedExperiment object of reference panel in **hg19** *for cell type deconvolution (which is used in* `decemedip` *function)*

Description

Function for assembling a SummarizedExperiment object of reference panel in **hg19** for cell type deconvolution (which is used in `decemedip` function)

Usage

```
makeReferencePanel(
  row_ranges,
  beta_matrix,
  cpg_coords,
  col_names = NULL,
  row_names = NULL,
  col_data = NULL,
  row_data = NULL
)
```

Arguments

row_ranges	A GRanges object that contains the genomic coordinates of *reference regions/sites*.
beta_matrix	A matrix that contains the beta values of reference regions. **Each row is a region and each column is a cell type.** If beta_matrix has row names or column names, the output SummarizedExperiment object will inherit them.
cpg_coords	A GRanges object that contains genomic coordinates of all CpGs in the genome. A ready-to-use CpG list for hg19 is available to download at https://github.com/nshen7/decemedip-experiments/blob/main/hg19.cpg.coords.rda. It is used for generating coloum n_cpg_100bp in the reference panel, which represents CpG density around the reference CpG.
col_names	An atomic vector of strings that indicates the column names, i.e., names of the cell types. Default is NULL. If not NULL, the column names of beta_matrix will be overwritten by this argument.
row_names	An atomic vector of strings that indicates the row names, i.e., names of the reference regions. Default is NULL. If not NULL, the row names of beta_matrix will be overwritten by this argument.
col_data	A DataFrame object that contains metadata for columns (i.e., cell types) if specified. Each row in col_data should contain info of a cell type in beta_matrix}. If input is a non-\co object, it will be converted to a DataFrame. Default is NULL.
row_data	A DataFrame object that contains metadata for row (i.e., reference regions) if specified. Each row in row_data should contain info of a reference region in beta_matrix}. If input is a non-\code{DataFrame object, it will be converted to a DataFrame. Default is NULL.

Value

An SummarizedExperiment object with each row represents a reference region and an assay named 'beta_matrix' that stores the beta values of reference regions.

Examples

```
row_ranges <- GenomicRanges::GRanges(
  seqnames = S4Vectors::Rle(c("chr1", "chr2", "chr3")),
  ranges = IRanges::IRanges(
    start = c(100, 200, 300),
    end = c(100, 200, 300)
  ),
  cpg_id = c("cpg_1", "cpg_2", "cpg_3") # CpG site IDs
)

cpg_coords <- GenomicRanges::GRanges(
  seqnames = S4Vectors::Rle(c("chr1", "chr1", "chr2", "chr2", "chr3", "chr3")),
  ranges = IRanges::IRanges(
    start = c(100, 101, 200, 201, 300, 301),
    end = c(100, 101, 200, 201, 300, 301)
  )
)
```

```
beta_matrix <- matrix(runif(6), nrow = 3)

makeReferencePanel(
  row_ranges = row_ranges,
  beta_matrix = beta_matrix,
  cpg_coords = cpg_coords
)
```

`pdx.counts.anc.se` *MeDIP-seq read counts on reference anchor CpGs of 3 PDX samples from Berchuck et al. 2022*

Description

This dataset represents a `SummarizedExperiment` object that contains MeDIP-seq read counts on reference anchor CpGs of 3 PDX samples from Berchuck et al. 2022. Each row is a CpG.

Usage

```
data(pdx.counts.anc.se)
```

Format

An object of class `SummarizedExperiment`.

Details

All coordinates are in hg19.

Examples

```
data(pdx.counts.anc.se)
pdx.counts.anc.se
```

`pdx.counts.cts.se` *MeDIP-seq read counts on reference cell type-specific CpGs of 3 PDX samples from Berchuck et al. 2022*

Description

This dataset represents a `SummarizedExperiment` object that contains MeDIP-seq read counts on reference cell type-specific CpGs of 3 PDX samples from Berchuck et al. 2022. Each row is a CpG.

Usage

```
data(pdx.counts.cts.se)
```

Format

An object of class SummarizedExperiment.

Details

All coordinates are in hg19.

Examples

```
data(pdx.counts.cts.se)
pdx.counts.cts.se
```

---

plotDiagnostics *Diagnostics for model fitting in* decemedip()

Description

Diagnostics for model fitting in decemedip()

Usage

```
plotDiagnostics(
  decemedip_output,
  plot_type,
  model_fit_n_samples = 100,
  model_fit_label_size = 12,
  model_fit_align = "hv",
  ...
)
```

Arguments

decemedip_output	The output from decemedip() function.
plot_type	A string value, either 'y_fit' or 'model_fit'. plot_type='y_fit' provides the fitted MeDIP-seq read counts vs. fractional methylation values, indicating the fitted relationship between MeDIP-seq counts and fractional methylation. plot_type='model_fit' provides a set of diagnostic plots for the fitted Stan model.
model_fit_n_samples	Number of randomly selected posterior samples for plotting the diagnostic plots of stan fit. For plot_type = 'model_fit' only.
model_fit_label_size	Label size in the plot grid. For plot_type = 'model_fit' only. See the argument label_size in cowplot::plot_grid for details.
model_fit_align	Specifies how graphs in the grid should be aligned. See the argument align in cowplot::plot_grid for details.

... Additional arguments to be fed into cowplot::plot_grid in the case of `plot_type = 'model_fit'`.

Value

An `ggplot` object.

Examples

```
data(pdx.counts.cts.se)
data(pdx.counts.anc.se)
# read counts of cell type-specific CpGs of the sample 'LuCaP_147CR'
counts_cts <- SummarizedExperiment::assays(pdx.counts.cts.se)$counts[, "LuCaP_147CR"]
# read counts of anchor CpGs of the sample 'LuCaP_147CR'
counts_anc <- SummarizedExperiment::assays(pdx.counts.anc.se)$counts[, "LuCaP_147CR"]

## The following function is commented due to Bioconductor's time constraints on package building
## Fit decemedip model
# output <- decemedip(counts_cts = counts_cts, counts_anc = counts_anc)
## Plot diagnostic plots
# plotDiagnostics(output, plot_type = "y_fit")
```

Index