Package 'GIMP'

July 18, 2025

Title Genomic Imprinting Methylation Patterns

Version 0.2.0

Description A package for analyzing Imprinting Control Regions (ICRs) DNA methylation. Supports both processed methylation data and raw IDAT files from Illumina arrays. Provides specialized tools for imprinting analysis including defect detection and interactive visualizations.

License MIT

Encoding UTF-8

LazyDataCompression xz

RoxygenNote 7.3.2

Depends R (>= 4.4.3), tidyverse, valr, reshape2, ggplot2, pheatmap, viridisLite, ggplotify, readr, shiny, plotly, BiocManager, minfi, doParallel

Imports shiny, shinydashboard, DT, plotly, minfi, IlluminaHumanMethylation450kanno.ilmn12.hg19, IlluminaHumanMethylationEPICanno.ilm10b4.hg19, IlluminaHumanMethylationEPICv2anno.20a1.hg38, readxl, tools, utils, stats

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

biocViews DNAMethylation, Microarray, Preprocessing, QualityControl, DifferentialMethylation, Epigenetics, MethylationArray

NeedsCompilation no

Contents

`bed450k` *BED 450K probes*

Description

This dataset contains the 450K array probes coordinates.

Usage

```
data(bed450k)
```

Format

BED file.

Examples

```
data(bed450k)
head(bed450k)
```

`bedEPICv1` *BED EPICv1 probes*

Description

This dataset contains the EPICv1 probes coordinates.

Usage

```
data(bedEPICv1)
```

Format

BED file.

Examples

```
data(bedEPICv1)
head(bedEPICv1)
```

bedEPICv2 *BED EPICv2 probes*

Description

This dataset contains the EPICv2 probes coordinates.

Usage

```
data(bedEPICv2)
```

Format

BED file.

Examples

```
data(bedEPICv2)
head(bedEPICv2)
```

calculate_detection_pvalues *Alternative Detection P-value Calculation*

Description

Alternative method for calculating detection p-values when standard minfi functions are not available.

Usage

```
calculate_detection_pvalues(rgSet)
```

Arguments

rgSet	RGChannelSet object from minfi

Value

Matrix of detection p-values or NULL if calculation fails

Examples

```
# This function requires actual IDAT data
# rgSet <- read.metharray.exp("path/to/idat/files")
# det_p <- calculate_detection_pvalues(rgSet)
```

check_minfi_functions *Check Available minfi Functions*

Description

Diagnostic function to check which minfi functions are available in the current installation.

Usage

```
check_minfi_functions()
```

Value

Character vector of all available minfi functions

Examples

```
# Check which minfi functions are available
available_funcs <- check_minfi_functions()
```

create_bedmeth *Create BED File Data from Methylation Array Annotations*

Description

This function generates a BED-format data frame from Illumina Human Methylation annotation files. The BED data includes chromosome, position, and probe ID information, and supports multiple annotation versions.

Usage

```
create_bedmeth(version = "v1")
```

Arguments

version	A character string specifying the annotation version to use. Options include '"450k"' for 450k array, '"v1"' for the EPIC version1 and '"v2"' for EPIC version2. Default is '"v1"'.

Value

A data frame in BED format containing columns:

chr	Chromosome name.
pos	Position on the chromosome.
probeID	Unique identifier for each probe.
end	End position, which is the same as 'pos' in this output.

Examples

```
# Create BED-format data with the default version (EPIC v1)
bed_data <- create_bedmeth()
head(bed_data)  # View the first few rows

# Use a different annotation version if available
bed_data_v2 <- create_bedmeth(version = "v2")
```

`create_sample_sheet_template` *Create Sample Sheet Template*

Description

Creates a template sample sheet for IDAT files to help users format their data correctly.

Usage

```
create_sample_sheet_template(
  sample_names = NULL,
  sentrix_ids = NULL,
  sentrix_positions = NULL,
  groups = NULL
)
```

Arguments

`sample_names`	Vector of sample names
`sentrix_ids`	Vector of Sentrix IDs (slide IDs)
`sentrix_positions`	Vector of Sentrix positions
`groups`	Optional vector of sample groups

Value

Data frame with sample sheet structure

Examples

```
# Create a basic template
template <- create_sample_sheet_template(
  sample_names = c("Sample1", "Sample2", "Sample3"),
  sentrix_ids = c("200123456", "200123456", "200123457"),
  sentrix_positions = c("R01C01", "R02C01", "R01C01"),
  groups = c("Control", "Control", "Case")
)
print(template)

# Create template with default values (minimal input)
simple_template <- create_sample_sheet_template(
  sample_names = c("Ctrl_01", "Case_01")
)
head(simple_template)
```

DMRs.hg19 *Imprinted Regions*

Description

This dataset contains the Human Imprinted regions coordinates in hg19.

Usage

```
data(DMRs.hg19)
```

Format

A data frame with iDMRs coordinates.

Examples

```
data(DMRs.hg19)
head(DMRs.hg19)
```

DMRs.hg38 *Imprinted Regions*

Description

This dataset contains the Human Imprinted regions coordinates in hg38.

Usage

```
data(DMRs.hg38)
```

Format

A data frame with iDMRs coordinates.

Examples

```
data(DMRs.hg38)
head(DMRs.hg38)
```

GIMP_app *Launch GIMP Shiny Application*

Description

Launches an interactive Shiny application for GIMP analysis. The app provides a graphical user interface for analyzing methylation patterns at Imprinted Control Regions (ICRs) without requiring R programming knowledge.

Usage

```
GIMP_app(max_upload_size_mb = 500)
```

Arguments

`max_upload_size_mb`
: Maximum file upload size in MB (default: 500)

Details

The GIMP Shiny app includes the following features:

- Upload methylation beta matrices from CSV files or raw IDAT files
- Analyze CpG coverage at ICRs
- Generate methylation heatmaps (beta, delta, and defect plots)
- Identify differentially methylated positions (DMPs)
- Explore specific ICR regions with interactive visualizations
- Export results and plots

Value

Opens the Shiny application in the default web browser. The function returns invisibly once the app is closed.

Examples

```
# Launch the GIMP Shiny app
GIMP_app()

# Launch with larger upload limit
GIMP_app(max_upload_size_mb = 1000)
```

ICRs_heatmap *Generate Heatmap of ICRs Methylation*

Description

This function generates a heatmap for visualizing methylation data of ICRs.

Usage

```
ICRs_heatmap(
  df_ICR,
  sampleInfo,
  control_label = "Control",
  case_label = "Case",
  bedmeth = "v1",
  order_by = "cord",
  annotation_col = NULL,
  plot_type = "beta",
  sd_threshold = 3
)
```

Arguments

df_ICR	A data frame or matrix containing methylation beta values for ICRs.
sampleInfo	A vector indicating the group labels (e.g., "Control" and "Case") for each sample in ‘df_ICR‘. Each element in ‘sampleInfo‘ should correspond to a sample in ‘df_ICR‘.
control_label	A character string specifying the label for the control group in ‘sampleInfo‘. Default is ‘"Control"‘.
case_label	A character string specifying the label for the case group in ‘sampleInfo‘. Default is ‘"Case"‘.
bedmeth	A character string specifying the BED data version for DMR coordinates. Options are ‘"v1"‘, ‘"v2"‘, or ‘"450k"‘. Default is ‘"v1"‘.
order_by	A character string specifying the ordering rows in the heatmap. Options are ‘"cord"‘ for coordinates or ‘"meth"‘ for methylation values. Default is ‘"cord"‘.
annotation_col	A named list of colors for each unique value in ‘sampleInfo‘. If ‘NULL‘, default colors are assigned using the "viridis" palette. Default is ‘NULL‘.
plot_type	A character string specifying the type of heatmap to generate. Options are ‘"beta"‘ for beta values, ‘"delta"‘ for values normalized against controls, and ‘"defect"‘ for defect matrix based on standard deviations. Default is ‘"beta"‘.
sd_threshold	A numeric value specifying the standard deviation threshold for detecting defects in the defect matrix. Only used if ‘plot_type‘ is ‘"defect"‘. Default is ‘3‘.

Value

A heatmap plot visualizing methylation of ICRs.

Examples

```
# Example sampleInfo with "Case" and "Control" labels for each sample
sampleInfo <- c(rep("Case", 10), rep("Control", 10))
ICRs_heatmap(df_ICR = my_ICR_data, sampleInfo = sampleInfo,
             annotation_col = list(Sample = c("darkgreen", "darkred")))
```

---

iDMPs *Identify Differentially Methylated Positions in ICRs*

Description

This function identifies differentially methylated positions (DMPs) between control and case groups using linear modeling and empirical Bayes methods.

Usage

```
iDMPs(data, sampleInfo, pValueCutoff = 0.05)
```

Arguments

`data`	A data frame containing CpG methylation data with annotation columns
`sampleInfo`	A factor or character vector indicating sample groups
`pValueCutoff`	P-value threshold for significance (default: 0.05)

Value

A list containing:

`fit`	Linear model fit object
`eBayesfit`	Empirical Bayes fit object
`topDMPs`	Data frame of significant DMPs
`allResults`	Data frame of all results
`groupLabels`	Group labels used in analysis

Examples

```
# Run DMP analysis
dmps <- iDMPs(data = ICRcpg, sampleInfo = sample_groups)
significant_dmps <- dmps$topDMPs
```

make_cpgs *Create ICR CpG Matrix*

Description

This function generates a CpG matrix for Imprinted Control Regions (ICR) using methylation data. The CpG matrix is constructed based on the provided BED data version.

Usage

```
make_cpgs(Bmatrix, bedmeth = "v1")
```

Arguments

Bmatrix	A data frame or matrix containing methylation beta values. Rows typically represent individual probes or CpGs, and columns represent samples.
bedmeth	A character string specifying the BED data version to use for CpG mapping. Options are '"v1"' (EPIC v1), '"v2"' (EPIC v2), or '"450k"' (450k array). Default is '"v1"'.

Value

A data frame representing the ICR CpG matrix, with rows as CpG probes and columns as samples.

Examples

```
# Create sample beta matrix for demonstration
set.seed(123)
n_probes <- 1000
n_samples <- 6

# Generate random probe IDs that might overlap with ICRs
sample_probes <- paste0("cg", sprintf("%08d", sample(1:50000000, n_probes)))
beta_matrix <- matrix(runif(n_probes * n_samples, 0.3, 0.8),
                      nrow = n_probes, ncol = n_samples)
rownames(beta_matrix) <- sample_probes
colnames(beta_matrix) <- paste0("Sample_", 1:n_samples)

# Generate the ICR CpG matrix with default BED version (EPIC v1)
ICRcpg <- make_cpgs(Bmatrix = beta_matrix, bedmeth = "v1")

# Use a different BED version, such as EPIC v2
ICRcpg_v2 <- make_cpgs(Bmatrix = beta_matrix, bedmeth = "v2")


# Simple usage with your own data:
# ICRcpg <- make_cpgs(Bmatrix = your_beta_matrix, bedmeth = "v1")
```

make_ICRs *Create the ICR Matrix*

Description

This function generates an ICR (Imprinted Control Region) matrix from a given beta matrix, using specified BED data for CpG mapping. The ICR matrix provides data organized by CpG probes and samples. The coordinates of the Human Imprinted regions are taken from https://doi.org/10.1080/15592294.2016.1264561

Usage

```
make_ICRs(Bmatrix, bedmeth = "v1")
```

Arguments

Bmatrix	A data frame or matrix containing methylation beta values. Rows should represent CpG probes, and columns represent samples.
bedmeth	A character string indicating the BED data version to use for CpG mapping. Options are '"v1"' (EPIC v1), '"v2"' (EPIC v2), or '"450k"' (450k array). Default is '"v1"'.

Value

A data frame representing the ICR matrix, structured by CpG probes and samples.

Examples

```
# Create sample beta matrix for demonstration
set.seed(123)
n_probes <- 1000
n_samples <- 6

# Generate random probe IDs that might overlap with ICRs
sample_probes <- paste0("cg", sprintf("%08d", sample(1:50000000, n_probes)))
beta_matrix <- matrix(runif(n_probes * n_samples, 0.3, 0.8),
                      nrow = n_probes, ncol = n_samples)
rownames(beta_matrix) <- sample_probes
colnames(beta_matrix) <- paste0("Sample_", 1:n_samples)

# Generate the ICR matrix with default BED version (EPIC v1)
ICRmatrix <- make_ICRs(Bmatrix = beta_matrix, bedmeth = "v1")


# Simple usage with your own data:
# ICRmatrix <- make_ICRs(Bmatrix = your_beta_matrix, bedmeth = "v1")
```

plot_cpgs_coverage *Plot ICR CpG Matrix with Counts and Percentage Coverage*

Description

This function plots the CpG coverage for Imprinted Control Regions (ICRs) using the provided data frame of CpG counts. It compares CpG counts in the specified BED data version for visual analysis and includes an additional plot for percentage coverage.

Usage

```
plot_cpgs_coverage(df_ICR_cpg, bedmeth = "v1")
```

Arguments

df_ICR_cpg	A data frame containing CpG counts for ICR regions. Each row represents a different CpG probe, and columns contain sample-related information.
bedmeth	A character string specifying the BED data version to use for mapping CpG coverage. Options are '"v1"' (EPIC v1), '"v2"' (EPIC v2), or '"450k"' (450k array). Default is '"v1"'.

Value

A list containing two plots (counts and percentage coverage) and the data frame with CpG counts and coverage information.

Examples

```
plot_cpgs_coverage(df_ICR_cpg_counts, bedmeth = "v1")
```

plot_line_ICR *Plot Line Plot for ICR Methylation*

Description

This function generates a line plot to visualize methylation values across a specified ICR. Users can choose between a static 'ggplot2' plot or an interactive 'plotly' plot.

Usage

```
plot_line_ICR(significantDMPs, ICRcpg, ICR, sampleInfo, interactive = TRUE)
```

Arguments

significantDMPs	A data frame containing information about significant DMPs. Must include columns 'ICR', 'start', and 'end'.
ICRcpg	A data frame or matrix containing CpG methylation data. Includes CpG coordinates ('cstart') and methylation values.
ICR	A character string specifying the name of the ICR region to be plotted.
sampleInfo	A character vector providing group labels (e.g., '"Control"' or '"Case"') for each sample in the methylation data.
interactive	A logical value indimessageing whether to return an interactive 'plotly' plot ('TRUE') or a static 'ggplot2' plot ('FALSE'). Default is 'TRUE'.

Value

A plot representing the line plot of methylation values across the specified ICR region, highlighting significant DMPs. The plot is either a 'ggplot2' object or a 'plotly' object, depending on the value of 'interactive'.

Examples

```
# Example data for significantDMPs
plot <- plot_line_ICR(significantDMPs, ICRcpg, ICR = "KCNQ1OT1:TSS-DMR", sampleInfo = sampleInfo, interactive
print(plot)
```

---

preview_idat_zip *Preview IDAT ZIP Contents*

Description

Preview the contents of an IDAT ZIP file without processing it. Useful for checking file structure before full processing.

Usage

```
preview_idat_zip(zip_file)
```

Arguments

zip_file	Path to ZIP file

Value

List with ZIP contents summary

Examples

```
# Preview IDAT ZIP file contents
# preview_info <- preview_idat_zip("methylation_data.zip")
# print(preview_info)
```

`read_idat_zip` *Read IDAT Files from ZIP Archive This function extracts and processes IDAT files from a ZIP archive containing IDAT files and a sample sheet, returning a beta value matrix ready for GIMP analysis.*

Description

Read IDAT Files from ZIP Archive This function extracts and processes IDAT files from a ZIP archive containing IDAT files and a sample sheet, returning a beta value matrix ready for GIMP analysis.

Usage

```
read_idat_zip(
  zip_file,
  sample_sheet_name = "samplesheet.csv",
  array_type = c("EPIC", "450k", "EPICv2"),
  temp_dir = NULL,
  normalize_method = c("quantile", "SWAN", "funnorm", "noob"),
  detection_pval = 0.01,
  remove_failed_samples = TRUE,
  n_cores = NULL
)
```

Arguments

`zip_file`	Path to ZIP file containing IDAT files and sample sheet
`sample_sheet_name`	Name of the sample sheet file in the ZIP (default: "samplesheet.csv")
`array_type`	Array type for annotation ("450k", "EPIC", "EPICv2")
`temp_dir`	Temporary directory for extraction (default: creates temporary directory)
`normalize_method`	Normalization method for minfi ("quantile", "SWAN", "funnorm", "noob")
`detection_pval`	P-value threshold for detection (default: 0.01)
`remove_failed_samples`	Remove samples with >10 percent failed probes (default: TRUE)
`n_cores`	Number of CPU cores to use for parallel processing (default: NULL for sequential processing)

Value

A list containing:

`beta_matrix`	Beta value matrix ready for GIMP analysis
`sample_info`	Sample information from the sample sheet
`qc_metrics`	Quality control metrics
`failed_samples`	Names of samples that failed QC

Examples

```
# Read IDAT files from ZIP
idat_data <- read_idat_zip("my_methylation_data.zip", array_type = "EPIC")
beta_matrix <- idat_data$beta_matrix

# Use parallel processing with 4 cores
idat_data <- read_idat_zip("my_methylation_data.zip", array_type = "EPIC", n_cores = 4)

# Use with GIMP functions
ICRcpg <- make_cpgs(Bmatrix = beta_matrix, bedmeth = "v1")
```

Index