Package 'GeneDMRs'

June 10, 2021

Type Package

Title Gene-based differentially methylated regions analysis

Version 1.0

Imports clusterProfiler, corrplot, dplyr, ffbase, genomation, KEGG.db, org.Hs.eg.db (for human), org.Mm.eg.db (for mouse), org.Ss.eg.db (for pig), org.Bt.eg.db (for cattle), pheatmap, plotrix, qqman, RCircos, VennDiagram

Author Xiao Wang, Dan Hao, Haja N. Kadarmideen.

Maintainer Xiao Wang <xiwa@dtu.dk> or <wangxiao880923@gmail.com>

Description GeneDMRs is an R package to detect the differentially methylated regions based on genes, gene body, CpG islands and gene body interacted with CpG island features. The output *coverage* file (i.e., *.bismark.cov*) of Bismark software for reduced-representation bisulfite sequencing (RRBS) can be directly used in the GeneDMRs package. Additionally, the methylation calling of whole genome bisulfite sequencing (WGBS) can be used if they are in the same format as the output *coverage* file of Bismark. *Bed* file (i.e., *.bed*) of refseq and cpgi can be directly used for the methylation levels in different gene or CpG island regions, and then filtered for the significant methylated genes or CpG islands. With the annotation of promoter, exon, intron, CpG island and CpG island shore based on R package genomation, gene body or CpG island feature regions and their interactive regions can also be analyzed by GeneDMRs package.

License GPL-3

RoxygenNote 6.0.1

R topics documented:

Index

Bedfile_read	*Read the standard bedfile of refseq or cpgi downloaded from UCSC*

Description

This function reads the *bed* file of refseq or cpgi and sorts them by chromosome and position. The dataset of the example are the mouse reference genes and CpG island information that are downloaded from UCSC website (http://genome.ucsc.edu/cgi-bin/hgTables). The R package genomation used here can divide the *refseq.bed* file into several gene body features, e.g., promoter, exon, intron regions and the *cpgi.bed* file into CpG island features, e.g., CpG island and CpG island shore.

refseq.bed file

```
chr1        134199214       134235457       NM_001291930    0       -       134202950
chr1        134199214       134234856       NM_001291928    0       -       134202950
chr1        134199214       134235457       NM_001282945    0       -       134202950
chr1        134199214       134235457       NM_001039510    0       -       134202950
chr1        134199214       134235457       NM_001008533    0       -       134202950
chr1        58713285 58733227 NM_009805       0       +       58726436 5873236
2   0       5       374,427,106,136,975,    0,13020,15770,17866,18967,
chr1        58713285 58758882 NM_207653       0       +       58726436 5875392
2   0       10      374,427,106,136,74,55,50,82,508,5099,   0,13020,15770,1
7866,18967,22372,26977,27679,39053,40498,
chr1        58711490 58758882 NM_001289704    0       +       58726436 5875392
2   0       10      102,427,106,136,74,55,50,82,508,5099,   0,14815,17565,1
9661,20762,24167,28772,29474,40848,42293,
```

```
chr1        8359738  9299877  NM_001290390     0        -        8363474  8803943  0        21
522422,639447,843440,938394,
chr1        8359738  9299877  NM_027671        0        -        8363474  8803943  0        21
522422,639447,843440,938394,
```

cpgi.bed file

```
chr1        84934572 84935054 CpG:_47
chr1        63176547 63177427 CpG:_78
chr1        125435174        125435976        CpG:_67
chr1        183368926        183369826        CpG:_93
chr1        3531624  3531843  CpG:_27
chr1        3670619  3671074  CpG:_34
chr1        3671654  3672156  CpG:_45
chr1        4491701  4493673  CpG:_165
chr1        4496947  4497608  CpG:_47
chr1        4571641  4572075  CpG:_44
```

Usage

```
Bedfile_read(paths = paste(system.file(package = "GeneDMRs"), "/methdata", sep=""), bedfile
= "refseq", suffix = ".txt", feature = FALSE, featurewrite = FALSE)
```

Arguments

paths	The path of bed file, with default the package path.
bedfile	The file name of *bed* file like “refseq” or “cpgi”. This file is downloaded from UCSC website, with default “refseq”.
suffix	The suffix of *bed* file, e.g., “.gz”, “.zip” and so on (some files are in text .txt format, then “.txt” or “.txt.gz”) , with default “.txt”.
feature	TRUE or FALSE value indicating whether to read the *bed* file with the features, with default FALSE. If feature = TRUE, the output of this function will contain the features e.g., promoter, exon, intron or CpG island, CpG island shore based on R package genomation.
featurewrite	TRUE or FALSE value indicating whether to write out the feature file to the given path, with default FALSE.

Value

A data frame contains four columns of chromosome, start position, end position. If feature = TRUE, the data frame is five columns with the added feature such as genebody or cpgfeature.

Reference

Akalin A, Franke V, Vlahovicek K, Mason C, Schubeler D (2014). “genomation: a toolkit to summarize, annotate and visualize genomic intervals.” *Bioinformatics*. doi: 10.1093/bioinformatics/btu775, http://bioinformatics.oxfordjournals.org/content/early/2014/12/04/bioinformatics.btu775.long.

Examples

```
inputrefseqfile <- Bedfile_read()
```

```
inputrefseqfile <- Bedfile_read(paths = paste(system.file(package = "GeneDMRs"), "/methdata",
sep=""), bedfile = "refseq", suffix = ".txt", feature = FALSE)
```

```
inputcpgifile <- Bedfile_read(paths = paste(system.file(package = "GeneDMRs"), "/methdata",
sep=""), bedfile = "cpgi", suffix = ".txt", feature = FALSE)

inputgenebodyfile <- Bedfile_read(bedfile = "refseq", feature = TRUE, featurewrite = TRUE)

inputcpgifeaturefile <- Bedfile_read(bedfile = "cpgi", feature = TRUE, featurewrite = FALSE)
```

Circos_plot *Plot the circos*

Description

This function outputs the circos plot for the methylation level and the density of gene, CpG island and CpG island shore on different chromosomes based on R RCircos package. All the files used in this function should contain chromosome, start position, and end position information that are required for R RCircos package.

Usage

```
Circos_plot(inputcytofile, inputmethfile_QC, inputrefseqfile, inputcpgifeaturefile,

labelname = regiongeneall_significant, linecolor = NULL)
```

Arguments

inputcytofile	The output of Cytofile_read() which contains the chromosome information.
inputmethfile_QC	The input file with methylation levels after quality control.
inputrefseqfile	The output of Bedfile_read() which contains the gene information.
inputcpgifeaturefile	The output of Bedfile_read() which contains the CpG island and CpG island shore information.
labelname	The label of gene names which could be the significant genes after Significant_filter(), with default regiongeneall_significant with differentially methylated genes. Sometimes, regiongenealls_significant will have some errors because it has unannotated chromosome name like chrUn_JH584304 or chrUn_NW_018084826v1. Thus, these chromosome names should be removed. If the labelname is from selfdefinedfile, then the file should contain the headers with chr (chromosome), start (start position), end (end position) and id (gene name).
linecolor	The colors of the lines plot for different methylation levels, with default NULL (black). If the linecolor is used, the length of colors should correspond to the length of groups.

Value

A circus figure with chromosomes, gene labels, the densities of the genes (track 3), CpG islands (track 4) and CpG island shores (track 5) and the methylation levels of different groups from the outermost circle to the innermost circle.

Reference

Hongen Zhang, Paul Meltzer, and Sean Davis. RCircos: an R package for Circos 2D track plots. BMC Bioinformatics, 2013, 14:244.

Examples

```
Circos_plot(inputcytofile, inputmethfile_QC, inputrefseqfile, inputcpgifeaturefile)
Circos_plot(inputcytofile, inputmethfile_QC, inputrefseqfile, inputcpgifeaturefile, labelname = selfdefinedfile, linecolor = c("blue", "orange", "green"))
```

Correlation_plot *Plot the methylation correlation*

Description

This function outputs the correlation plot for the methylation level of different samples or groups based on R package corrplot.

Usage

```
Correlation_plot(inputmethfile_QC, unmeth_exclude = TRUE)
```

Arguments

inputmethfile_QC	The input file with methylation levels, with default inputmethfile after quality control.
unmeth_exclude	TRUE or FALSE value indicating whether to exclude the unmethylated sites or regions, with default TRUE

Value

A correlation figure.

Examples

```
Correlation_plot(inputmethfile_QC)

Correlation_plot(siteall)

Correlation_plot(regiongenealls)

Correlation_plot(genefeatureall_cpgfeature)

Correlation_plot(genefeatureall_cpgfeature, unmeth_exclude = FALSE)
```

Chomosome_pieplot *Pie plot based on different chromosomes*

Description

This function outputs the pie plot for the percentages of sites or regions in different chromosomes.

Usage

```
Chromosome_pieplot(genefeatureall_cpgfeature_significantcpgisland,
genefeatureall_cpgfeature_significantshore = NULL, methydirection = "both", title = "Pie plot
for chromosome")
```

Arguments

genefeatureall_cpgfeature_significantcpgisland	The input file with chromosomes, which can be files with/without Significant_filter().
genefeatureall_cpgfeature_significantshore	Another input file with chromosomes, e.g., genefeatureall_cpgfeature_significant shore file for comparison, with default NULL.
methydirection	The methylation direction when the input file contains the methylation difference column i.e., methdiff after Logic_regression(), which can be “hypo”, “hyper” and “both”, with the default “both” for both directions.
title	Figure titles, with the default “Pie plot for chromosome”.

Value

A pie figure in different chromosomes.

Examples

```
Chromosome_pieplot(genefeatureall_cpgfeature_significantcpgisland, title = "")

Chromosome_pieplot(genefeatureall_cpgfeature_significantcpgisland, title = "CpGisland")

Chromosome_pieplot(genefeatureall_cpgfeature_significantcpgisland,
genefeatureall_cpgfeature_significantshore = genefeatureall_cpgfeature_significantshore, title
= c("CpGisland","Shore"))

Chromosome_pieplot(siteall, title = "All cytosine sites") # Only consider the annotated
chromosomes and the unannotated chromosomes will be discarded #

Chromosome_pieplot(siteall_Qvalue, title = "All cytosine sites")

Chromosome_pieplot(siteall_significant, title = "Significant cytosine sites")

Chromosome_pieplot(siteall_Qvalue, methydirection = "hyper", title = "Hyper-methylated
distribution"))
```

```
Chromosome_pieplot(siteall_significant, methydirection = "hypo", title = "Hypo-methylated pie plot"))

Chromosome_pieplot(regiongeneall_Qvalue, methdirection = "hyper", title = "Hyper-methylated genes")
```

Cytofile_read *Read the cyto file*

Description

This function reads the chromosome information from *cyto* file (*cytoBandIdeo.txt*) and sort them by chromosome and position. The dataset of the example is the mouse genome information downloaded from UCSC website (http://hgdownload.cse.ucsc.edu/goldenPath/mm10/database/cytoBandIdeo.txt.gz).

Usage

```
Cytofile_read(paths = paste(system.file(package = "GeneDMRs"), "/methdata", sep=""), cytofile = "cytoBandIdeo", suffix = ".txt.gz")
```

Arguments

paths	The path of input file, with default the package path.
cytofile	The name of input *cyto* file that is downloaded from UCSC website, with default "cytoBandIdeo".
suffix	The suffix of input *cyto* file, e.g., ".gz", ".zip" and so on (some files are in text .txt format, then ".txt" or ".txt.gz"), with default ".txt.gz".

Value

A data frame contains chromosome, start position, end position.

Examples

```
inputcytofile <- Cytofile_read()

inputcytofile <- Cytofile_read(paths = paste(system.file(package = "GeneDMRs"), "/methdata", sep=""), cytofile = "cytoBandIdeo", suffix = ".txt.gz")
```

DMC_feature *Annotate the differentially methylated cytosine (DMC) to features*

Description

This function annotates the differentially methylated cytosine (DMC) after statistical test Logic_regression() to gene body or CpG island features.

Usage

```
DMC_feature(siteall_significant, featureid = TRUE, featurefile = list(inputgenebodyfile,
inputcpgifeaturefile))
```

Arguments

siteall_significant	The input file with DMC sites.
featureid	TRUE or FALSE value indicating whether to include the feature id or not, with the default TRUE. The feature id will output the id of first file of the featurefile list e.g., the id of inputgenebodyfile.
featurefile	The input feature files e.g., inputgenebodyfile and inputcpgifeaturefile, with default two files in a list as featurefile = list(inputgenebodyfile, inputcpgifeaturefile), and it can also be one file without a list e.g., featurefile = inputgenebodyfile.

Value

A data frame contains DMC sites with features.

Examples

```
siteall_significant_feature <- DMC_feature(siteall_significant, featurefile =
list(inputgenebodyfile, inputcpgifeaturefile))

siteall_significant_feature <- DMC_feature(siteall_significant, featureid = FALSE, featurefile =
list(inputgenebodyfile, inputcpgifeaturefile))

siteall_significant_feature <- DMC_feature(siteall_significant, featureid = TRUE, featurefile =
inputgenebodyfile)
```

DMC_methfile_QC — *Merge the methylation file after quality control with DMCs*

Description

This function merges the methylation file after quality control of all samples with the DMCs after Significant_filter().

Usage

```
DMC_methfile_QC(inputmethfile_QC, siteall_significant)
```

Arguments

inputmethfile_QC	Input methylation file after quality control.
siteall_significant	Input DMCs file.

Value

A data frame by merging two input files of inputmethfile_QC and siteall_significant.

Examples

```
DMC_inputmethfile_QC <- DMC_methfile_QC(inputmethfile_QC, siteall_significant)
```

Enrich_plot *Enrich plot for GO terms and pathways*

Description

This function outputs the groups, GO terms and pathways plot for the enrichment based on R package clusterProfiler, org.Mm.eg.db (Mouse for example) and KEGG.db.

Usage

```
Enrich_plot(regiongenealls_significant, adjustpvaluecut = 0.1, enrichterm = "pathway",
Dbannotation = "org.Mm.eg.db",  keggorganism = "mmu", listnum = 20, title = "Enrichment for
significant gene", expressionfile_significant = NULL, expressionfile_genetype = NULL)
```

Arguments

regiongenealls_significant	Differentially methylated genes.
adjustpvaluecut	Threshold of the adjusted P values for the enrichment, with default 0.1.
enrichterm	The term need to be analyzed, which can be “GOgroup”, “GO”, “pathway”, with default “pathway”.
category	TRUE or FALSE value indicating whether to divide the enrichments into two categories, i.e., hypo/hyper methylated or down/up regulated, with default TRUE.
Dbannotation	Annotation dataset, with default “org.Mm.eg.db” of mouse.
keggorganism	Species name for KEGG enrichment, with default “mmu” of mouse.
listnum	The list of display number, with default 20.
title	The title of figure, with default “Enrichments for significant gene”.
expressionfile_significant	An additional file for differentially expressed genes, which includes gene name and Log fold change (LogFC). This is an optional file for GO terms and pathways, with default NULL.
expressionfile_genetype	The gene type of expressionfile_significant file, which can be “REFSEQ”, “ENTREZID”, “SYMBOL”, or other gene types that can be used in clusterProfiler, with default NULL.

Value

A dot-plot figure of enrichment.

Examples

```
Enrich_plot(regiongenealls_significant, enrichterm = "GOgroup", Dbannotation =
"org.Mm.eg.db", title = "Biological process for significant gene")

Enrich_plot(regiongenealls_significant, enrichterm = "GO", Dbannotation = "org.Mm.eg.db",
title = "Go term for significant gene")

Enrich_plot(regiongenealls_significant, enrichterm = "GO", Dbannotation = "org.Hs.eg.db", title
= "Go term for significant gene") # for human data #

Enrich_plot(regiongenealls_significant, adjustpvaluecut = 0.2, enrichterm = "pathway",
Dbannotation = "org.Mm.eg.db", keggorganism = "mmu", title = "Pathway for significant gene")

Enrich_plot(regiongenealls_significant, enrichterm = "pathway", category = FALSE,
keggorganism = "hsa", Dbannotation = "org.Hs.eg.db", title = "Pathway for significant gene") #
for human data #


expressionfile_significant <- read.table(paste(system.file(package = "GeneDMRs"),
"/methdata/DEgenes.txt", sep=""), header = T) # read DEgene file #

Enrich_plot(regiongenealls_significant, adjustpvaluecut = 0.2, enrichterm = "GO",
Dbannotation = "org.Mm.eg.db", title = "Go term for significant gene in two categories",
expressionfile_significant = expressionfile_significant, expressionfile_genetype = "SYMBOL")

Enrich_plot(regiongenealls_significant, enrichterm = "pathway", Dbannotation =
"org.Mm.eg.db", keggorganism = "mmu", title = "Pathway for significant gene in two
categories", expressionfile_significant = expressionfile_significant, expressionfile_genetype =
"SYMBOL")
```

Feature_pieplot *Pie plot based on different features*

Description

This function outputs the pie plot of feature percentages in gene body or CpG island mainly for DMC sites with features.

Usage

```
Feature_pieplot(siteall_significant_feature, methdirection = "both", title = "Pie plot for
feature", threeDplot = TRUE)
```

Arguments

siteall_significant_feature	The input file with features, mainly for DMC sites with features.
methydirection	The methylation direction when the input file contains the

	methylation difference column i.e., methdiff after Logic_regression(), which can be “hypo”, “hyper” and “both”, with the default “both” for both directions.
title	Figure titles, with the default “Pie plot for chromosome”.
threeDplot	TRUE or FALSE value indicating whether to pie plot in three dimensions based on R pacakge plotrix, with the default TRUE.

Value

A pie figure in different features.

Examples

```
Feature_pieplot(siteall_significant_feature)

Feature_pieplot(siteall_significant_feature, methdirection = "hypo")

Feature_pieplot(siteall_significant_feature, title = c("Gene body", "CpG island"))

Feature_pieplot(siteall_significant_feature, title = c("Pie plot for Gene body", "Pie plot for CpG island"), threeDplot = FALSE)

Feature_pieplot(siteall_significant_feature, methdirection = "hyper", title = c("Pie plot for Gene body", "Pie plot for CpG island"))
```

GeneDMRs — *Gene-based differentially methylated regions analysis (GeneDMRs) and install the dependencies*

Description

GeneDMRs is an R package to detect the differentially methylated regions based on genes (DMG), gene body (DMP, DME, DMI), CpG islands and gene body interacted with CpG island features (e.g., DMG/DMP/DME/DMI_CpG island and DMG/DMP/DME/DMI_CpG island shore). This function can install the other R packages for the dependencies of GeneDMRs.

Usage

```
GeneDMRs(Dbannotation = "org.Mm.eg.db")
```

Arguments

Dbannotation	The annotation dataset for enrichment, with default "org.Mm.eg.db" of mouse.

Value

A list of required R packages.

Examples

```
GeneDMRs(Dbannotation = "org.Mm.eg.db")
```

Group_boxplot *Boxplot the methylation levels for groups*

Description

This function outputs the methylation levels of all the groups in boxplot without considering other features.

Usage

```
Group_boxplot(regiongeneall, ttest = TRUE, title = "Group boxplot", col = NULL)
```

Arguments

regiongeneall	The input file with group methylation levels.
ttest	TRUE or FALSE value indicating whether to perform the Student t-test, with default TRUE.
title	The figure title, with default “Group boxplot among genes”.
col	The boxplot colors, with default NULL.

Value

A boxplot figure with groups.

Examples

```
Group_boxplot(regiongeneall)

Group_boxplot(genebodypromoterall, title = "Three groups among genes in promoter region")

Group_boxplot(regiongeneall, ttest = FALSE, title = "Three groups among genes", col = c("red",
"green", "blue"))
```

Group_cpgfeature_boxplot *Boxplot the methylation levels for groups based on CpG island features*

Description

This function outputs the methylation levels in boxplot for one or more groups based on CpG island features, e.g., CpG island and CpG island shore features.

Usage

```
Group_cpgfeature_boxplot(genefeatureall_cpgfeature, groupnum = "all", ttest = TRUE,
cpgfeaturelable = c("CpGisland", "Shore"), title = c("Group1", "Group2", "Group3"), col = NULL)
```

Arguments

genefeatureall_cpgfeature	The input file with group methylation levels and CpG island features.
groupnum	Group number, with default "all" for all of the groups.
ttest	TRUE or FALSE value indicating whether to perform the Student t-test, with default TRUE.
cpgfeaturelable	CpG island features, with default "CpGisland" and "Shore". Only one CpG island feature can also be available, e.g., "CpGisland".
title	The figure title, with default "Group1", "Group2" and "Group3".
col	The boxplot colors, with default NULL.

Value

A boxplot figure with groups and CpG island features.

Examples

```
Group_cpgfeature_boxplot(genefeatureall_cpgfeature, groupnum = 1)

Group_cpgfeature_boxplot(genefeatureall_cpgfeature, groupnum = "all", ttest = TRUE,
cpgfeaturelable = c("CpGisland", "Shore"), title = c("Group1", "Group2", "Group3"), col =
c("blue", "red", "green"))
```

Genebody_cpgfeature_boxplot — *Boxplot the methylation levels for gene body based on CpG island features*

Description

This function outputs the methylation levels in boxplot for one or more features of gene body based on CpG island features, e.g., CpG island and CpG island shore features.

Usage

```
Genebody_cpgfeature_boxplot(genefeatureall_cpgfeature, genebodyname =
c("promoters","exons","introns","TSSes"), ttest = TRUE, cpgfeaturelable = c("CpGisland",
"Shore"), title = c("Promoter", "Exon", "Intron", "TSS"), col = NULL)
```

Arguments

genefeatureall_cpgfeature	The input file of methylation levels with gene body and CpG island features.
genebodyname	The name of gene body features e.g., promoter, exon, intron and TSSes, with default "promoters", "exons",

	"introns", "TSSes".
ttest	TRUE or FALSE value indicating whether to perform the Student t-test, with default TRUE.
cpgfeaturelable	CpG island features, with default "CpGisland" and "Shore".
title	The figure title, with default "Promoter", "Exon", "Intron" and "TSS".
col	The boxplot colors, with default NULL.

Value

A boxplot figure with gene body and CpG island features.

Examples

```
Genebody_cpgfeature_boxplot(genefeatureall_cpgfeature)

Genebody_cpgfeature_boxplot(genefeatureall_cpgfeature, genebodyname =
    c("promoters","exons"), ttest = TRUE, cpgfeaturelable = c("CpGisland", "Shore"), title =
    c("Promoter", "Exon"), col = c("blue", "red"))

Genebody_cpgfeature_boxplot(genefeatureall_cpgfeature, genebodyname =
    c("promoters","exons","introns","TSSes"), ttest = TRUE, cpgfeaturelable = c("CpGisland",
    "Shore"), title = c("Promoters", "Exons", "Introns", "TSSes"), col = c("blue", "red", "green",
    "purple"))
```

Heatmap_plot *Heat map plot for chromosomes and features*

Description

This function outputs the heat map plot for methylation level in different chromosomes of differentially methylated genes with features based on R package pheatmap.

Usage

```
Heatmap_plot(regiongeneall_significant, featurename = NULL, title = "Methylation level",
display_numbers = FALSE, number_format = "%.0f", cluster_rows = FALSE, cluster_cols = TRUE,
gaps_row = c(1,2), gaps_col = NULL)
```

Arguments

regiongeneall_significant	The input file of methylation levels with differentially methylated genes or the genes in different gene body features.
featurename	Feature name of the output file from Significant_filter() for genefeatureall_cpgfeature file, that is "CpGisland" or "Shore", with default NULL.
title	The figure title, with the default "Methylation level".
display_numbers	TRUE or FALSE value indicating whether to display the

	methylation value in the figure, with default FALSE.
number_format	The displayed number of the methylation value in round format.
cluster_rows	TRUE or FALSE value indicating whether to cluster the row, with the default FALSE.
cluster_cols	TRUE or FALSE value indicating whether to cluster the column, with the default TRUE.
gaps_row	TRUE or FALSE value indicating whether to divide the row, with the default c(1,2) that divide the rows into three parts by row 1 and row 2.
gaps_col	TRUE or FALSE value indicating whether to divide the column, with the NULL.

Value

A heat map figure with methylation levels.

Examples

```
Heatmap_plot(regiongeneall_significant)

Heatmap_plot(genefeatureall_cpgfeature_significantcpgisland, featurename = "CpGisland",
display_numbers = FALSE, title = "Methylation level (%) for genes with CpG island")

Heatmap_plot(genefeatureall_cpgfeature_significantshore, featurename = "Shore", title =
"Methylation level (%) for genes with shores")

Heatmap_plot(genefeatureall_cpgfeature_significantshore, featurename = "Shore", title =
"Methylation level (%) for genes with shores", cluster_cols = FALSE)

Heatmap_plot(regiongeneall_significant, title = "Methylation level (%) for genes",
display_numbers = FALSE)

Heatmap_plot(regiongeneall_significant, title = "Methylation level (%) for genes",
display_numbers = FALSE, cluster_rows = TRUE, gaps_row = NULL)
```

Logic_regression	*Logistical regression analysis for each region or each cytosine site*

Description

This function tests each region or each cytosine site by logistical regression model to achieve the P values and then be adjusted to Q values to account for multiple hypothesis testing.

Usage

```
Logic_regression(genefeatureall_cpgfeature, covariates = NULL, adjustedmethod = "fdr",
diffgroup = NULL)
```

Arguments

genefeatureall_cpgfeature	The input file with methylation levels to be tested.
covariates	Extra covariates used in the model, with the default NULL.
adjustedmethod	The methods to adjust P values to Q values, with the default "fdr" method. The adjustedmethod could be "holm", "Hochberg", "hommel", "bonferroni", "BH", "BY", "none" methods as well.
diffgroup	Methylation difference between two groups, with the default NULL, that is the max group - min group. The two groups can be manually selected e.g., diffgroup = c("group1", "group2").

Value

A data frame of region gene or region cpgi or those regions with different features or cytosine sites, by accompanying with P values, Q values and methylation differences.

Examples

```
regiongeneall_Qvalue <- Logic_regression(regiongeneall)

regiongenealls_Qvalue <- Logic_regression(regiongenealls)

regioncpgall_Qvalue <- Logic_regression(regioncpgall , adjustedmethod = "fdr")

regiongenebodyall_Qvalue <- Logic_regression(regiongenebodyall, diffgroup = c("group1",
"group2"))

regiongeneall_cpgfeature_Qvalue <- Logic_regression(regiongeneall_cpgfeature)

genefeatureall_cpgfeature_Qvalue <- Logic_regression(genefeatureall_cpgfeature)

genefeatureall_Qvalue <- Logic_regression(genefeatureall, adjustedmethod = "bonferroni")


siteall_Qvalue <- Logic_regression(siteall, adjustedmethod = "fdr") # for each cytosine site #

siteall_Qvalue <- Logic_regression(siteall, adjustedmethod = "fdr", diffgroup = c("group1",
"group2"))
```

Manhattan_plot *Manhattan plot for all cytosines or regions*

Description

This function outputs the Manhattan plot for all cytosines or regions in different chromosomes with significant line based on R package qqman.

Usage

```
Manhattan_plot(siteall_Qvalue, chrlabs = NULL, col = c("black", "grey"), ylab = "-log(Q value)",
suggestiveline = -log10(1e-02), genomewideline = -log10(1e-03))
```

Arguments

siteall_Qvalue	The input file with Q values from DMR_test(), e.g., regiongeneall_Qvalue, genefeatureall_cpgfeature_Qvalue or others with Q values.
chrlabs	The label of chromosomes, with default NULL.
col	The color of plots, with default black and grey.
suggestiveline	The significant line, with default 0.01.
genomewideline	The genome-wide significant line, with default 0.001.

Value

A Manhattan figure with Q values.

Examples

```
Manhattan_plot(siteall_Qvalue, ylab = "-log(Q-value)")

Manhattan_plot(regiongenealls_Qvalue, chrlabs = c(1:18,"X"), col = c("green","orange"), genomewideline = -log10(1e-02))

Manhattan_plot(genefeatureall_cpgfeature_Qvalue, ylab = c("-log(Q value) for CpG island", "-log(Q value) for Shore"), col = c("red","blue"), suggestiveline = -log10(5e-02), genomewideline = -log10(1e-02))
```

Methfile_read — *Read the methylation file*

Description

This function reads all of the methylation files and generates one file with all samples including methylated read coverages (Cs) and unmethylated read coverages (Ts). It can automatically test how many samples and how many replicates in each group and the distribute them from 1_1, 1_2 to the final file by headers. The methylation files should be the standard *coverage* file (i.e., *.bismark.cov*) outputted from Bismark software. The dataset of the example is the Reduced representation bisulfite sequencing (RRBS) data of DNA methylation for mouse myeloid progenitor tissue from GEO (Accession number: GSE62392) (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE62392).

1_1.gz after rename the *coverage* file (i.e., *.bismark.cov*) of the first replicate of first group

```
chr1 3020877 3020877  97.46835 77  2
chr1 3020891 3020891  92.40506 73  6
chr1 3020946 3020946  88.67925 47  6
chr1 3020988 3020988  98.64865 73  1
chr1 3021013 3021013 100.00000 74  0
chr1 3094122 3094122   0.00000  0  1
chr1 3094126 3094126 100.00000  1  0
chr1 3150008 3150008 100.00000  3  0
chr1 3150022 3150022 100.00000  3  0
chr1 3150068 3150068 100.00000  1  0
```

Usage

```
Methfile_read(paths = paste(system.file(package = "GeneDMRs"), "/methdata", sep=""), suffix
= ".gz")
```

Arguments

paths	The path of methylation file, with default the package path.
control_paths	The path of control groups, with default NULL.
case_paths	The path of case groups, with default NULL.
suffix	The suffix of methylation file, e.g., ".gz", ".zip" and so on (some files are in text .txt format, then ".txt" or ".txt.gz"), with default ".gz".

Value

A data frame contain chromosome, position, and Cs & Ts for different replicates and groups.

Examples

```
inputmethfile <- Methfile_read()

inputmethfile <- Methfile_read(paths = paste(system.file(package = "GeneDMRs"),
"/methdata", sep=""), suffix = ".gz")


# if only case and control group (n = 2) paths are provided #

controls <- c("C:/Users/GeneDMRs/methdata/1_1.gz", C:/Users/GeneDMRs/methdata/1_2.gz",
"C:/Users/GeneDMRs/methdata/1_3.gz")

cases <- c("C:/Users/GeneDMRs/methdata/2_1.gz", "C:/Users/GeneDMRs/methdata/2_1.gz")

inputmethfile <- Methfile_read(control_paths = controls, case_paths = cases)


# inputmethfile
```

#chr	posi	Cs1_1	Ts1_1	Cs1_2	Ts1_2	Cs1_3	Ts1_3	Cs2_1	Ts2_1	Cs2_2	Ts2_2
chr1	3020877	77	2	77	7	49	2	31	4	68	0
chr1	3020891	73	6	78	6	49	2	33	2	68	0
chr1	3020946	47	6	96	17	71	9	52	5	71	12
chr1	3020988	73	1	58	0	57	6	55	2	61	2
chr1	3021013	74	0	56	2	59	4	49	8	63	0
chr1	3531651	11	1	25	0	13	1	7	0	15	1
chr1	3531658	12	0	25	0	12	2	7	0	16	0
chr1	3531671	12	0	25	0	13	1	6	1	16	0
chr1	3531676	12	0	25	0	14	0	7	0	16	0
chr1	3531680	12	0	22	3	10	3	7	0	14	1

Methfile_QC — *Quality control for the input methylation file*

Description

This function discards the cytosine sites with low read coverage (quantile) or high read coverage (quantile).

Usage

```
Methfile_QC(inputmethfile, low_coveragenum = 10, high_coveragenum = NULL, low_quantile = NULL, high_quantile = 99.9, samplenum_QC = "all")
```

Arguments

inputmethfile	The input of methylation file after Methfile_read().
low_coveragenum	The minimum read coverage to be discarded, with default 10.
high_coveragenum	The maximum read coverage to be discarded, with default NULL.
low_quantile	The minimum quantile of read coverage to be discarded, with default NULL.
high_quantile	The maximum quantile of read coverage to be discarded, with default 99.99.
samplenum_QC	The sample numbers under quanlity control (e.g., samplenum_QC = 3 means that if three of five samples at one cytosine site have unqualified read coverage, then this site will be discarded), with default "all" samples.

Value

A data frame contain chromosome, position, and Cs & Ts for different replicates and groups after quality control.

Examples

```
inputmethfile_QC <- Methfile_QC(inputmethfile)

inputmethfile_QC <- Methfile_QC(inputmethfile, low_coveragenum = 20, high_quantile = 99.99)

inputmethfile_QC <- Methfile_QC(inputmethfile, low_coveragenum = 10, high_coveragenum = 100, samplenum_QC = 3)
```

Methmean_region *Calculate the methylation mean for regions*

Description

This function outputs the methylation mean for different groups based on gene and CpG island regions by matching with cytosine. It is also for gene body of promoter, exon, intron and TSSes regions, cgpi feature of CpG island and CpG island shores and their interactive regions e.g., promoter CpG island.

Usage

Methmean_region(inputmethfile_QC, inputrefseqfile, cpgifeaturefile = NULL, chrnum = "all", posistart = NULL, posiend = NULL, featureid = NULL, featurename = NULL)

Arguments

inputmethfile_QC	The input of methylation file after quality control.
inputrefseqfile	The input file with regions e.g., inputrefseqfile/inputcpgifile with 4 columns or inputgenebodyfile/inputcpgifeaturefile with 5 columns.
cpgifeaturefile	The input of CpG island feature file e.g., inputcpgifeaturefile, with default NULL. If provided, the output file is methylation mean of inputrefseqfile or inputgenebodyfile with CpG island and CpG island shore features.
chrnum	The chromosome number or all chromosomes (all) or all chromosomes with unannotated sites (alls), with default “all”.
posistart	Start position if requested, with default NULL.
posiend	End position if requested, with default NULL.
featureid	NCBI ID of specific gene or all the genes, with default NULL. The CpG id can also be used like “cpgi1” or “shore2”.
featurename	Different gene body features of promoter, exon, intron and TSSes. The CpG island features can also be used that are “CpGisland” and “Shores”.

Value

A data frame of the methylation mean of provided regions with/without different features.

Examples

```
Methmean_region(inputmethfile_QC, inputrefseqfile, chrnum = "alls", featureid = c("NM_001244353", "NM_001244864")) # find sepecific gene by NCBI ID #

Methmean_region(inputmethfile_QC, inputrefseqfile, chrnum = "chr1", posistart = 21800, posiend = 21900)

regiongenechr <- Methmean_region(inputmethfile_QC, inputrefseqfile, chrnum = c("chr1","chr2"))

regiongeneall <- Methmean_region(inputmethfile_QC, inputrefseqfile, chrnum = "all")

DMC_regiongeneall <- Methmean_region(DMC_inputmethfile_QC, inputrefseqfile, chrnum = "all") # Calculate DMC first and then recalculate the methylation mean by replacing the RRBS cytosine sites #

regiongenealls <- Methmean_region(inputmethfile_QC, inputrefseqfile, chrnum = "alls") # alls include unannotated CpG site like chrUn_NW_018084826v1 #

Methmean_region(inputmethfile_QC,inputcpgifile,"chr1", 21800, 21900) # acturally regiongenepart = regioncpgpart #

regioncpgchr <- Methmean_region(inputmethfile_QC, inputcpgifile, chrnum = c("chr1","chr2"))

regioncpgall <- Methmean_region(inputmethfile_QC, inputcpgifile, chrnum = "all")
```

```
regioncpgalls <- Methmean_region(inputmethfile_QC, inputcpgifile, chrnum = "alls")


regiongenebodychr <- Methmean_region(inputmethfile_QC, inputgenebodyfile, chrnum =
c("chr1","chr2"))

regiongenebodyall <- Methmean_region(inputmethfile_QC, inputgenebodyfile, chrnum = "all")

regiongenebodyalls <- Methmean_region(inputmethfile_QC, inputgenebodyfile, chrnum =
"alls")

regioncpgifeaturechr <- Methmean_region(inputmethfile_QC, inputcpgifeaturefile, chrnum =
c("chr1","chr2"))

regioncpgifeatureall <- Methmean_region(inputmethfile_QC, inputcpgifeaturefile, chrnum =
"all")

regioncpgifeaturealls <- Methmean_region(inputmethfile_QC, inputcpgifeaturefile, chrnum =
"alls")


partgenebody <- Methmean_region(inputmethfile_QC, inputgenebodyfile, featureid =
"NM_001244353")

partgenebodyexon <- Methmean_region(inputmethfile_QC, inputgenebodyfile, featureid =
"NM_001244353", featurename = "exons")

partgenebodyall <- Methmean_region(inputmethfile_QC, inputgenebodyfile, featureid =
"NM_001244353", featurename = c("promoters","exons","introns","TSSes"))

genebodypromoterall <- Methmean_region(inputmethfile_QC, inputgenebodyfile, featureid =
"all", featurename = "promoters")

genebodyexonall <- Methmean_region(inputmethfile_QC, inputgenebodyfile, featureid = "all",
featurename = "exons")

genefeatureall <- Methmean_region(inputmethfile_QC, inputgenebodyfile, featureid = "all",
featurename = c("promoters","exons","introns","TSSes")) #long time #

partcpgi <- Methmean_region(inputmethfile_QC, inputcpgifeaturefile, featureid = "cpgi1")

partshore <- Methmean_region(inputmethfile_QC, inputcpgifeaturefile, featureid = "shore10")

cpgislandall <- Methmean_region(inputmethfile_QC, inputcpgifeaturefile, featureid = "all",
featurename = "CpGisland")

cpgshoreall <- Methmean_region(inputmethfile_QC, inputcpgifeaturefile, featureid = "all",
featurename = "Shores") #long time #

cpgfeatureall <- Methmean_region(inputmethfile_QC, inputcpgifeaturefile, featureid = "all",
featurename = c("CpGisland", "Shores") #long time #


genebodychr_promoter <- Methmean_region(inputmethfile_QC, inputgenebodyfile, chrnum =
"chr1", featureid = "all", featurename = "promoters")
```

```
cpgchr_island <- Methmean_region(inputmethfile_QC, inputcpgifeaturefile, chrnum = "chr1",
featureid = "all", featurename = "CpGisland")


# when the cpgifeaturefile = inputcpgifeaturefile is provided #

regiongenechr_cpgfeature <- Methmean_region(inputmethfile_QC, inputrefseqfile,
cpgifeaturefile = inputcpgifeaturefile, chrnum = c("chr1","chr2"))

regiongeneall_cpgfeature <- Methmean_region(inputmethfile_QC, inputrefseqfile,
cpgifeaturefile = inputcpgifeaturefile, chrnum = "all")

regiongenealls_cpgfeature <- Methmean_region(inputmethfile_QC, inputrefseqfile,
cpgifeaturefile = inputcpgifeaturefile, chrnum = "alls")

genebodypromoterall_cpgfeature <- Methmean_region(inputmethfile_QC, inputgenebodyfile,
cpgifeaturefile = inputcpgifeaturefile, featureid = "all", featurename = "promoters")

genebodyexonall_cpgfeature <- Methmean_region(inputmethfile_QC, inputgenebodyfile,
cpgifeaturefile = inputcpgifeaturefile, featureid = "all", featurename = "exons")

genefeatureall_cpgfeature <- Methmean_region(inputmethfile_QC, inputgenebodyfile,
cpgifeaturefile = inputcpgifeaturefile, featureid = "all", featurename =
c("promoters","exons","introns","TSSes")) #long time #


# windows #

windowfileall <- Methmean_region(inputmethfile_QC, windowfile, chrnum = "all")

windowfilealls <- Methmean_region(inputmethfile_QC, windowfile, chrnum = "alls")
```

Methmean_site	*Calculate the methylation mean for cytosine sites*

Description

This function outputs the methylation mean for each cytosine site. It will calculate methylation difference along each group.

Usage

```
Methmean_site(inputmethfile_QC)
```

Arguments

inputmethfile_QC	The input of methylation file after quality control.

Value

A data frame of the methylation mean of provided cytosine sites.

Examples

siteall <- Methmean_site(inputmethfile_QC)

Quick_DMCs — *Quick use the GeneDMRs package for differentially methylated cytosine sites*

Description

This function outputs the differentially methylated cytosine sites (DMCs).

Usage

Quick_DMCs(paths = paste(system.file(package = "GeneDMRs"), "/methdata", sep=""), suffixmeth = ".gz")

Arguments

paths	The path of input file, with default the package path.
control_paths	The path of control groups, with default NULL.
case_paths	The path of case groups, with default NULL.
suffixmeth	The suffix of methylation file, e.g., ".gz", ".zip" and so on (some files are in text .txt format, then ".txt" or ".txt.gz"), with default ".gz".

Value

Outputs DMC results.

Examples

allDMCs <- Quick_DMCs()

if only case and control group (n = 2) paths are provided

controls <- c("C:/Users/GeneDMRs/methdata/1_1.gz", C:/Users/GeneDMRs/methdata/1_2.gz", "C:/Users/GeneDMRs/methdata/1_3.gz")

cases <- c("C:/Users/GeneDMRs/methdata/2_1.gz", "C:/Users/GeneDMRs/methdata/2_1.gz")

allDMCs <- Quick_DMCs(control_paths = controls, case_paths = cases)

Quick_GeneDMRs — *Quick use the GeneDMRs package for gene based differentially methlated regions*

Description

This function outputs a series of results and figures for gene based regions' methylation analysis.

Usage

```
Quick_GeneDMRs(paths = paste(system.file(package = "GeneDMRs"), "/methdata", sep=""),
suffixmeth = ".gz", bedfile = "refseq", suffixbed = ".txt", Dbannotation = "org.Mm.eg.db",
keggorganism = "mmu")
```

Arguments

paths	The path of input file, with default the package path.
control_paths	The path of control groups, with default NULL.
case_paths	The path of case groups, with default NULL.
suffixmeth	The suffix of methylation file, e.g., ".gz", ".zip" and so on (some files are in text .txt format, then ".txt" or ".txt.gz"), with default ".gz".
bedfile	The file name of bed file for "refseq". This file is downloaded from UCSC website, with default "refseq".
suffixbed	The suffix of bed file, e.g., ".gz", ".zip" and so on (some files are in text .txt format, then ".txt" or ".txt.gz"), with default ".txt".
Dbannotation	The annotation dataset for enrichment, with default "org.Mm.eg.db" of mouse.
keggorganism	The species name for KEGG enrichment, with default "mmu" of mouse.

Value

Outputs a series of DMG results.

Examples

```
allDMGs <- Quick_GeneDMRs()

allDMGs_mouse <- Quick_GeneDMRs(Dbannotation = "org.Mm.eg.db", keggorganism =
"mmu")


# if only case and control group (n = 2) paths are provided #

controls <- c("C:/Users/GeneDMRs/methdata/1_1.gz", C:/Users/GeneDMRs/methdata/1_2.gz",
"C:/Users/GeneDMRs/methdata/1_3.gz")

cases <- c("C:/Users/GeneDMRs/methdata/2_1.gz", "C:/Users/GeneDMRs/methdata/2_1.gz")

allDMGs <- Quick_GeneDMRs(paths = "C:/Users/GeneDMRs/methdata", control_paths =
controls, case_paths = cases)
```

Sample_boxplot	*Boxplot the methylation levels or read numbers in different samples*

Description

This function outputs the methylation levels or read numbers of the selected genes or all the genes in the different samples.

Usage

```
Sample_boxplot(inputmethfile, inputrefseqfile, Meth_plot = TRUE, ylab = "Methylation level", refseqname = NULL, col = NULL)
```

Arguments

inputmethfile	The input file with methylation levels.
inputrefseqfile	The input of gene regions.
Meth_plot	TRUE or FALSE value indicating whether to plot the methylation levels, with default TRUE, otherwise to plot the read numbers.
ylab	The label of y axis, with default “Methylation level”.
refseqname	NCBI ID of specific gene, with default NULL.
col	The boxplot colors, with default NULL.

Value

A boxplot figure with all the samples.

Examples

```
Sample_boxplot(inputmethfile_QC, inputrefseqfile)

Sample_boxplot(inputmethfile_QC, inputrefseqfile, refseqname = "NM_001244864")

Sample_boxplot(inputmethfile_QC, inputrefseqfile, refseqname = c("NM_001244864", "NM_001244534"))

Sample_boxplot(inputmethfile_QC, inputrefseqfile, , ylab = "Methylation level (%)", refseqname = c("NM_001244864", "NM_001143697", "NM_213902"), col = c("red", "green", "blue"))

Sample_boxplot(inputmethfile_QC, inputrefseqfile, Meth_plot = FALSE, ylab = "Read number", col = c("red", "blue"))

Sample_boxplot(inputmethfile_QC, inputrefseqfile, Meth_plot = FALSE, ylab = "Read number", refseqname = c("NM_001244864", "NM_001244534"))

Sample_boxplot(inputmethfile_QC, inputrefseqfile, Meth_plot = FALSE, ylab = "Read number", refseqname = c("NM_001244864", "NM_001143697", "NM_213902"), col = c("red", "green", "blue"))
```

Significant_filter *Filter the significant regions or cytosine sites*

Description

This function filters significant regions or cytosine sites based on Q value and methylation difference.

Usage

```
Significant_filter(genefeatureall_cpgfeature_Qvalue, qvalue = 0.01, methdiff = 0, featureout = 1)
```

Arguments

genefeatureall_cpgfeature_Qvalue	The input file with Q values and methylation differences need to be filtered.
qvalue	Threshold of Q values that Q values larger than this will be discarded, with default 0.01.
methdiff	Threshold of methylation differences that methylation differences less than this will be discarded, with the default 0.
featureout	Which feature will be filtered, with default 1. When featureout = 2, it means that the second feature will be filtered and outputted.

Value

A data frame of the significant regions or cytosine sites.

Examples

```
genefeatureall_cpgfeature_significantcpgisland <- Significant_filter(genefeatureall_cpgfeature_Qvalue)

genefeatureall_cpgfeature_significantshore <- Significant_filter(genefeatureall_cpgfeature_Qvalue, featureout = 2)

genefeatureall_cpgfeature_significantcpgisland <- Significant_filter(genefeatureall_cpgfeature_Qvalue, qvalue = 0.001, methdiff = 0.01, featureout = 1)

regiongeneall_cpgfeature_significantcpgisland <- Significant_filter(regiongeneall_cpgfeature_Qvalue, methdiff = 0.05, featureout = 1)

regiongeneall_significant <- Significant_filter(regiongeneall_Qvalue)

regiongenealls_significant <- Significant_filter(regiongenealls_Qvalue, methdiff = 0.1)

siteall_significant <- Significant_filter(siteall_Qvalue)

siteall_significant <- Significant_filter(siteall_Qvalue, qvalue = 0.001, methdiff = 0.1)
```

Venn_plot	*Venn plot for the common CpG island and CpG island shore*

Description

This function outputs the venn plot for the common CpG island and CpG island shore regions that are covered by methylated cytosine sites based on R package VennDiagram.

Usage

```
Venn_plot(genefeatureall_cpgfeature_Qvalue, title = "Venn plot", fillcolor =
c("cornflowerblue","green"))
```

Arguments

genefeatureall_cpgfeature_Qvalue	The input file with two features.
title	Figure title, with default “Venn plot”.
fillcolor	Filled color, with default “cornflowerblue” and “green”

Value

A venn figure in two features.

Examples

```
Venn_plot(genefeatureall_cpgfeature)

Venn_plot(genefeatureall_cpgfeature_Qvalue)

Venn_plot(genefeatureall_cpgfeature_Qvalue, fillcolor = c("red","blue"))
```

Volcano_plot *Volcano plot for all the cytosines*

Description

This function outputs the volcano plot for all the cytosines with Q values and methylation differences.

Usage

```
Volcano_plot(siteall_Qvalue, title = "Volcano for Q value and methylation difference", qvalue =
0.01, methdiffpercentage = c(5, 10, 15, 20, 15), pointcolor = c("red", "purple", "orange",
"yellow", "blue", "green"))
```

Arguments

siteall_Qvalue	The input file with Q values and methylation differences.
title	Figure title, with default “Volcano for Q value and methylation

	difference".
qvalue	Threshold of Q values that Q values less than this will be colored, with default 0.01.
methdiffpercentage	Threshold of methylation level (%) differences that methylation differences larger than this will be colored, with default 5, 10, 15, 20, 25.
pointcolor	Point plot color, with default "red", "purple", "orange", "yellow", "blue", "green".

Value

A volcano figure.

Examples

```
Volcano_plot(siteall_Qvalue)

Volcano_plot(siteall_Qvalue, pointcolor = c("red", "blue", "yellow", "purple", "orange",
"green"))

Volcano_plot(siteall_Qvalue, title = "Volcano plot", qvalue = 0.001, methdiffpercentage = c(10,
15, 20, 30, 40), pointcolor = c("red", "purple", "orange", "yellow", "blue", "green"))
```

Window_divide *Divide the genome to windows*

Description

This function outputs the window regions of the whole genome.

Usage

```
Window_divide(inputcytofile, windowbp = 1000000)
```

Arguments

inputcytofile	The input *cyto* file with chromosome information.
windowbp	Window length in base pair (bp) to be divided, with default 1,000,000.

Value

A data frame with window regions.

Examples

```
windowfile <- Window_divide(inputcytofile)

windowfile <- Window_divide(inputcytofile, windowbp = 10000)
```