Package 'MLExpResso'

August 27, 2017

Title Integrative analyses and visualization of gene expression and DNA methylation data

Version 0.2.0.0

Description MLExpResso is an R package for integrative analyses and visualization of gene expression and DNA methylation data. Key functions of this package are: identification of genes with affected expression, identification of DMR - differentially methylated regions, identification of regions with changes in expression and methylation, visualization of identified regions. The joint modeling and visualization of genes expression and methylation improve interpretability of identified signals.

Depends R (>= 3.0.0)

License GPL

Encoding UTF-8

LazyData true

Imports AnnotationDbi,
DESeq,
DESeq2,
dplyr,
edgeR,
ggplot2,
ggrepel,
ggthemes,
grid,
gridExtra,
kableExtra,
limma,
methyAnalysis,
org.Hs.eg.db,
reshape,
reshape2,
roxygen2,
scales,
stringr,
TxDb.Hsapiens.UCSC.hg18.knownGene,
utils,

RoxygenNote 6.0.1

Suggests knitr,
rmarkdown,
testthat

VignetteBuilder knitr

R topics documented:

`aggregate_probes` *Aggregating CpG probes to genes*

Description

Function `aggregate_probes` aggregates CpG probes to corresponding genes.

Usage

```
aggregate_probes(data, keep = NULL,
  genom.data = MLExpResso::illumina_humanmethylation_27_data,
  genes.col = 11, probes.col = 1)
```

Arguments

`data`	Data frame containing methylation values for CpG probes. Columns correspond to probes, rows to samples.
`keep`	The name of the column or vector of columns names we want to keep.
`genom.data`	Data frame which contains information about CpG probes and corresponding genes, by default in our package we use `illumina_humanmethylation_27_data`.
`genes.col`	Number of column in genom.data containing informations about genes (genes symbols).
`probes.col`	Number of column in genom.data containing informations about probes (probes symbols).

Value

A data frame with CpG probes aggregated to genes. If there were more than one probe corresponding to a gene, value is a mean of those probes.

Examples

```
## Not run:
library(MLExpRessoData)
BRCA_genes <- aggregate_probes(BRCA_methylation_all, keep="SUBTYPE")

## End(Not run)
```

`calculate_comparison_table`

Results from test for two datasets.

Description

Function `calculate_comparison_table` produces a dataset containing p-values and folds from tests evaluated on two datasets e.g methylation or expression. In addition, it produces an importance ranking column, which is the geometric mean of p-values from both tests and a column with a number of probes related to the gene.

Usage

```
calculate_comparison_table(data1, data2, condition1, condition2, test1, test2,
  genom.data = MLExpResso::illumina_humanmethylation_27_data,
  genes.col = 11)
```

Arguments

`data1`	First dataset, suitable for `test1`.
`data2`	Second dataset, suitable for `test2`.
`condition1`	Condition for first dataset.
`condition2`	Condition for second dataset.
`test1`	Type of test for first dataset.
`test2`	Type of test for second dataset.
`genom.data`	Data frame which contains information about CpG probes and corresponding genes, by default in our package we use `illumina_humanmethylation_27_data`.
`genes.col`	Number of column in genom.data containing informations about genes (genes symbols).

Value

Data frame containing logatithm of fold and p-values from chosen tests.

See Also

`calculate_test`

Examples

```
## Not run:
library(MLExpRessoData)
condition_exp <- ifelse(BRCA_exp$SUBTYPE == "LumA","LumA", "other")
condition_met <- ifelse(BRCA_met$SUBTYPE == "LumA","LumA", "other")

BRCA_methylation_gen <- aggregate_probes(BRCA_met)

data_met <- BRCA_methylation_gen
data_exp <- BRCA_exp
compare <- calculate_comparison_table(data_exp,data_met,cond_exp,cond_met, "nbinom2", "ttest")

## End(Not run)
```

`calculate_test` *Statistical computations for methylation and expression data.*

Description

Function `calculate_test` computes log folds and p-values for choosen test for data from methylation or expression. The function uses: t-test, negative binomial test, likelihood-ratio test(LRT), quasi-likelihood F-test(QLF). By default function calls the `t-test`.

Usage

```
calculate_test(data, condition, test = "ttest", ...)
```

Arguments

`data`	An object of the class appropriate for the given test. More in `details` section.
`condition`	Factor of levels coresponding to order of samples in data.
`test`	Variable defining test. Values: ttest, nbinom, nbinom2, lrt, qlf, methyanalysis. More in `details` section.
`...`	Other parameters e.g. ‘adjust.method‘ for argument ‘ttest‘.

Details

Each test may require different data. In this section we will describe details for each availible test:

ttest Student's t-test
Test for expression and methylation.
This test is based on function `lmFit` from `limma` package.
For this test you should aggregate CpG probes to genes using function: `aggregate_probes`.
Data for this test should have following structure: columns correspond to genes, rows to samples.

nbinom Negative binomial test
Test for expression.
This test is based on function `nbinomTest` from `DESeq` package.
Data for this test should have following structure: columns correspond to genes, rows to samples.
Calculations may take some time. It is suggested to use `nbinom2` parameter.

nbinom2 Negative binomial test
Test for expression.
This test is based on function DESeq from DESeq2 package.
Data for this test should have following structure: columns correspond to genes, rows to samples.

lrt Likelihood-ratio test (LRT)
Test for expression.
This test is based on function glmFit and glmLRT from edgeR package.
Data for this test should have following structure: columns correspond to genes, rows to samples.

qlf Quasi-likelihood F-test (QLF)
Test for expression.
This test is based on functions glmQLFit and glmQLFTest from edgeR package.
Data for this test should have following structure: columns correspond to genes, rows to samples.

methyanalysis Slide window smoothing
Test for methylation.
This test is based on function detectDMR.slideWin from methyAnalysis package. It requires a special class of argument data - MethyGenoSet.

Value

A data frame with the following columns:

id	The id of the observable, taken from the row names of the counts slots.
log2.fold	The log2 of the fold change.
pval	The p-values for rejecting the null hypothesis about the means equality.
mean	Column correspond to means for each gene defined by condition and mean for all probes.

See Also

calculate_comparison_table, aggregate_probes

Examples

```
## Not run:

library(MLExpRessoData)
BRCA_methylation_gene <- aggregate_probes(BRCA_methylation_all)

condition_m <- ifelse(BRCA_methylation_all$SUBTYPE == "LumA", "LumA", "other")
test_methylation <- calculate_test(BRCA_methylation_gene, condition_m, "ttest")


## End(Not run)
```

generate_report *Generate report for MLExpResso package.*

Description

Function generate_report produces a pdf report containing information about functionalities of package MLExpResso.

Usage

```
generate_report(data.m, data.e, condition.m, condition.e, test.m, test.e, genes)
```

Arguments

data.m	Data frame containing information for methylation.
data.e	Data frame containing information for expression.
condition.m	Condition for methylation.
condition.e	Condition for expression.
test.m	Test results for methylation.
test.e	Test results for expression.
genes	The name of the gene or vector of genes names we want to visualise.

Value

PDF with report containing information about MLExpResso package.

See Also

calculate_test, aggregate_probes

Examples

```
## Not run:
library(MLExpRessoData)
cond_met <- ifelse(BRCA_met$SUBTYPE == "LumA","LumA","other")
cond_exp <- ifelse(BRCA_exp$SUBTYPE == "LumA","LumA","other")

BRCA_methylation_gene <- aggregate_probes(BRCA_met, keep = "SUBTYPE")

test1  <- calculate_test(BRCA_exp[,-1], condition_exp, test = "lrt")
test2  <- calculate_test(BRCA_methylation_gene[,-1], condition_met, test = "ttest")
data_met <- BRCA_met
data_exp <- BRCA_exp

report <- generate_report(data_exp, data_met, cond_exp, cond_met, test1, test2, "BRCA2")


## End(Not run)
```

illumina_humanmethylation_27_data
Illumina_humanmethylation_27_data

Description

The variables are as follows:

- Name. CG number from CG database (format cg########).
- GenomeBuild. Genome build.
- Chr. Chromosome on which the target locus is located.
- MapInfo. Cancer subtype.
- Source. Genomic position of C in CG dinucleotide.
- SourceVersion. Genomic position source.
- SourceSeq. Original sequence of the region covered by assay probes.
- TSS_Coordinat. Transcription start site genomic coordinate.
- Gene_Strand. Gene strand.
- Gene_ID. RefSeq identifier (GeneID).
- Symbol. Gene Symbol.
- Synonym. Gene synonyms.
- Accession. Gene Accession (this is the accession of the longest transcript).
- GID. GI ID.
- Annotation. Gene annotation from NCBI database.
- Product. Gene product description from NCBI database.
- Distance_to_TSS. Distance of CG dinucleotide to transcription start site.
- CPG_ISLAND. oolean variable denoting whether the loci is located in a CpG island (by relaxed definition) .
- CPG_ISLAND_LOCATIONS. Chromosomal location and genomic coordonates of the CpG island from NCBI database.
- MIR_CPG_ISLAND. Chromosome:start-end of upstream CPG island from a micro RNA.
- MIR_NAMES. Name of micro RNA near locus.

Format

data.frame

`plot_diff_boxplot` *Boxplot for expression in groups*

Description

Function `plot_diff_boxplot` generates a boxplot of values from choosen data frame column with division in groups.

Usage

```
plot_diff_boxplot(data, condition = "", gene, sqrt.trans = FALSE,
  title = TRUE)
```

Arguments

`data`	Data frame containing interesing values.
`condition`	Vector of length equal to numer of rows of data. Vector `condition` should contains names of groups corresponding to rows.
`gene`	String containing name of gene with values for boxplot.
`sqrt.trans`	Root square y-axis transformation.
`title`	Plot title, by default id of gene.

Value

Object of class ggplot containing boxplot for values from choosen gene broken down to groups.

See Also

`plot_gene`

Examples

```
## Not run:
library(MLExpRessoData)
condition_expr <- ifelse(BRCA_exp7$SUBTYPE == "LumA","LumA","other")
plot_diff_boxplot(BRCA_exp, "BRCA2", condition_expr, sqrt.trans=TRUE)

## End(Not run)
```

plot_gene *Visualisations for genes.*

Description

Function plot_gene generates a dashboard with methylation_path for methylation and boxplots for groups.

Usage

```
plot_gene(data.m, data.e, condition.m, condition.e, gene, ...)
```

Arguments

data.m	Data frame containing information from methylation.
data.e	Data frame containing information from expression.
condition.m	Condition for methylation data (data.m).
condition.e	Condition for expression data (data.e).
gene	Gene name.
...	Optional arguments (e.g from plot_methylation_path or plot_diff_boxplot functions)

Value

Object of class ggplot containing visualisation of methylation on gene and boxplot for values from choosen gene broken down to groups.

See Also

plot_methylation_path, plot_diff_boxplot

plot_methylation_path, plot_diff_boxplot

Examples

```
## Not run:
library(MLExpRessoData)
condition_exp <- ifelse(BRCA_exp$SUBTYPE == "LumA","LumA","other")
condition_met <- ifelse(BRCA_met$SUBTYPE == "LumA","LumA","other")
plot_gene(BRCA_met,BRCA_exp,condition_met, condition_exp, "ICAM2")

## End(Not run)
```

plot_methylation_path *Methylation path on chosen gene.*

Description

Function plot_methylation_path visualizes a chosen gene with marked CpG probes. Y axis describes methylation level. X axis describes a location of the probe on the chromosome. Horizontal lines show the mean methylation level for each Island in a division to groups. Groups are defined by colors. Large dots symbolize means of methylation level for CpG probes, small dots symbolize methylation levels for each observation. Also, we can exact the line corresponding to a gene. In this case, we see what are the locations of probes on a gene in HG18 coordinates. We can as well draw locations of CpG islands.

Usage

```
plot_methylation_path(data, condition, gene, show.gene = FALSE,
  observ = FALSE, islands = TRUE, title = TRUE, ...)
```

Arguments

data	Data frame containing values from methylation: columns corespond to CpG probes, rows to samples.
condition	Vector of levels corresponding to order of samples in data.
gene	Name of chosen gene.
show.gene	Logical. If TRUE line corresopnding to gene will be drawn.
observ	Logical. If TRUE dots corresponding to CpG probes will be drawn.
islands	Logical. If TRUE line corresopnding to islands should be drawn.
title	Logical. If TRUE title saying what gene we visualise will be added.
...	Other parameters.

Value

Object of class ggplot containing visualisation of methylation on gene.

See Also

plot_gene

Examples

```
## Not run:
library(MLExpRessoData)
condition_met <- ifelse(BRCA_met$SUBTYPE == "LumA","LumA", "other")
plot_methylation_path(BRCA_met, condition_met, "BRCA2")

## End(Not run)
```

plot_volcano *Visualise the p-values of expression and methylation for genes.*

Description

Function plot_volcano draws a plot with p-values and fold logarithm from methylation or expression when we use the t-test.

Usage

```
plot_volcano(data, line = NA, names = NA, ylog = TRUE, ngen = NA,
  title = NA, fold_line = NA, values = FALSE)
```

Arguments

data	Data frame containing result of chosen test.
line	P-value on which we draw a line.
names	P-value below which we want to draw genes names.
ylog	Logical. If TRUE values on y-axis will be logarithmized.
ngen	Character symbol or vector of gene names.
title	Character containing title for plot.
fold_line	Value on which we want to draw a vertical line on both sides of zero.
values	Logical. If TRUE p-values and log fold for chosen gene will be add to a plot. By default we use FALSE.

Value

An object of class ggplot containing volcano plot of p-values versus log2.fold for each gene.

See Also

plot_volcanoes

Examples

```
## Not run:
library(MLExpRessoData)
plot_volcano(BRCA_met, values=TRUE)

## End(Not run)
```

plot_volcanoes *Visulisations for methylation and expression.*

Description

Function plot_volcanoes generate a dashboard with volcano plots for expression and methylation. Also it adds a tables with basic statistics.

Usage

```
plot_volcanoes(data.m, data.e, condition.m, condition.e, gene = NA,
  test.m = list(), test.e = list(), values = FALSE, font.size = 6)
```

Arguments

data.m	Data frame containing information for methylation.
data.e	Data frame containing information for expression.
condition.m	Condition for methylation data (data.m).
condition.e	Condition for expression methylation (data.e).
gene	Gene name.
test.m	List of tests results for methylation.
test.e	List of tests results for expression
values	Logical. If TRUE p-values and log fold for chosen gene will be added to a plot. By default we use FALSE.
font.size	Font size in stats table.

Value

An object of class ggplot containing volcano plots of p-values versus log2.fold for gene for chosen number of tests. Also there are added simple statistisc about chosen gene.

See Also

plot_volcano

plot_volcano

Examples

```
## Not run:
library(MLExpRessoData)
cond_exp <- ifelse(BRCA_exp$SUBTYPE == "LumA","LumA","other")
cond_met <- ifelse(BRCA_met$SUBTYPE == "LumA","LumA","other")

BRCA_met_gen <- aggregate_probes(BRCA_met[,-1])

test1  <- calculate_test(BRCA_exp[,-1], condition_exp, test="lrt")
test2  <- calculate_test(BRCA_met_gen, condition_met, test="ttest")

plot_volcanoes(BRCA_met[,-1],BRCA_exp[,-1],cond_met, cond_exp, "ICAM2", test1, test2, values=TRUE)

## End(Not run)
```

Index