Package 'MAGeCKFlute'

October 16, 2019

Type Package

Title Integrative analysis pipeline for pooled CRISPR functional genetic screens

Version 1.4.3

Date 2019-8-6

Author Wubing Zhang, Feizhen Wu, Binbin Wang

Maintainer Wubing Zhang<Watson5bZhang@gmail.com>

Description MAGeCKFlute is designed to surporting downstream analysis, utilizing the gene summary data provided through MAGeCK or MAGeCK-VISPR. Quality control, normalization, and screen hit identification for CRISPR screen data are performed in pipeline. Identified hits within the pipeline are categorized based on experimental design, and are subsequently interpreted by functional enrichment analysis.

License GPL (>=3)

VignetteBuilder knitr

Depends R (>= 3.5), ggplot2, stats, grDevices, utils, gridExtra

Suggests knitr, rmarkdown, BiocStyle, org.Mm.eg.db

Imports ggExtra, ggsci, ggrepel, clusterProfiler, png, data.table, pheatmap, sva, DOSE, biomaRt, grid, pathview

LazyData TRUE

NeedsCompilation no

biocViews Software, FunctionalGenomics, CRISPR, ImmunoOncology, BatchEffect, QualityControl, Normalization, GeneSetEnrichment, Pathways, Visualization

RoxygenNote 6.1.1

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

git_branch RELEASE_3_9

git_last_commit 73044ca

git_last_commit_date 2019-08-08

Date/Publication 2019-10-15

R topics documented:

`arrangePathview` *Kegg pathway view and arrange grobs on page*

Description

Kegg pathway view and arrange grobs on page.

Usage

```
arrangePathview(genelist, pathways = c(), top = 4, ncol = 2,
  title = "Group A", sub = "Negative control normalized",
  organism = "hsa", view_allpath = FALSE, output = ".",
  path.archive = ".", kegg.native = TRUE)
```

Arguments

`genelist`	a data frame with columns of ENTREZID, Control and Treatment. The columns of Control and Treatment represent gene score in Control and Treatment sample.
`pathways`	character vector, the KEGG pathway ID(s), usually 5 digit, may also include the 3 letter KEGG species code.
`top`	integer, specifying how many top enriched pathways to be visualized.
`ncol`	integer, specifying how many column of figures to be arranged in each page.
`title`	optional string, or grob.
`sub`	optional string, or grob.
`organism`	character, either the kegg code, scientific name or the common name of the target species. This applies to both pathway and gene.data or cpd.data. When KEGG ortholog pathway is considered, species="ko". Default species="hsa", it is equivalent to use either "Homo sapiens" (scientific name) or "human" (common name).
`view_allpath`	boolean, specifying whether view all pathways. Default view_allpath='FALSE', and only plot top enriched pathways.
`output`	Path to save plot to.
`path.archive`	character, the directory of KEGG pathway data file (.xml) and image file (.png). Users may supply their own data files in the same format and naming convention of KEGG's (species code + pathway id, e.g. hsa04110.xml, hsa04110.png etc) in this directory. Default kegg.dir="." (current working directory).
`kegg.native`	logical, whether to render pathway graph as native KEGG graph (.png) or using graphviz layout engine (.pdf). Default kegg.native=TRUE.

Value

plot on the current device

Author(s)

Wubing Zhang

See Also

KeggPathwayView

Examples

```
data(mle.gene_summary)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(mle.gene_summary)
colnames(dd)[2:3] = c("Control", "Treatment")
arrangePathview(dd, "hsa00534", title=NULL, sub=NULL, organism="hsa")
```

BatchRemove *Batch effect removal*

Description

Batch effect removal

Usage

```
BatchRemove(mat, batchMat, log2trans = FALSE)
```

Arguments

mat	A data frame, each row is a gene, and each column is a sample.
batchMat	A data frame, the first column should be 'Samples'(matched colnames of mat) and the second column is 'Batch'. The remaining columns could be Covariates.
log2trans	Boolean, specifying whether do logarithmic transformation before batch removal.

Value

A list contrains two objects, including data and p.

Author(s)

Wubing Zhang

See Also

ComBat

Examples

```
edata = matrix(c(rnorm(2000, 5), rnorm(2000, 8)), 1000)
colnames(edata) = paste0("s", 1:4)
batchMat = data.frame(sample = colnames(edata), batch = rep(1:2, each = 2))
edata1 = BatchRemove(edata, batchMat)
print(edata1$p)
```

CellCycleView *Visualize the estimate cell cycle compared to control.*

Description

Estimate cell cycle time in different samples by linear fitting of beta scores.

Usage

```
CellCycleView(beta, ctrlname, treatname, main = NULL, filename = NULL,
  width = 5, height = 4, ...)
```

Arguments

beta	Data frame, which has columns of ctrlname and other samples.
ctrlname	A character, specifying the names of control samples.
treatname	A character, specifying the name of treatment samples.
main	As in 'plot'.
filename	Figure file name to create on disk. Default filename="NULL", which means no output.
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in ggsave.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

Examples

```
data(mle.gene_summary)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(mle.gene_summary)
CellCycleView(dd, ctrlname = "dmso", treatname = "plx")
```

countsummary *Count summary data generated by running MAGeCK count*

Description

The summary of QC values at count level

Usage

```
data("countsummary")
```

Format

A data frame with 4 observations on 13 variables.

References

https://www.ncbi.nlm.nih.gov/pubmed/25494202 https://www.ncbi.nlm.nih.gov/pubmed/25476604

Examples

```
data("countsummary")
head(countsummary)
```

CutoffCalling *Quantile of normal distribution.*

Description

Compute cutoff from a normal-distributed vector.

Usage

```
CutoffCalling(d, scale = 1)
```

Arguments

d	A numeric vector.
scale	Boolean or numeric, specifying how many standard deviation will be used as cutoff.

Value

A numeric value.

Examples

```
CutoffCalling(rnorm(10000))
```

DensityDiffView *Density plot*

Description

Plot the density of beta score deviations.

Usage

```
DensityDiffView(beta, ctrlname = "Control", treatname = "Treatment",
  main = NULL, filename = NULL, width = 5, height = 4, ...)
```

Arguments

beta	Data frame, including ctrlname and treatname as columns.
ctrlname	A character, specifying the name of control sample.
treatname	A character, specifying the name of treatment sample.
main	As in 'plot'.
filename	Figure file name to create on disk. Default filename="NULL", which means no output.
width	As in ggsave.
height	As in ggsave.
...	Other parameters in ggsave.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

Examples

```
data(mle.gene_summary)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(mle.gene_summary)
# Density plot of beta score deviation between control and treatment
DensityDiffView(dd, ctrlname = "dmso", treatname = "plx")
```

DensityView *Density plot for gene beta scores in Control and Treatment*

Description

Plot the density of gene beta scores in two samples.

Usage

```
DensityView(beta, samples = NULL, main = NULL, xlab = "Beta Score",
  filename = NULL, width = 5, height = 4, ...)
```

Arguments

beta	Data frame, including samples as columns.
samples	Character, specifying sample names in beta.
main	As in 'plot'.
xlab	As in 'plot'.
filename	Figure file name to create on disk. Default filename="NULL", which means don't save the figure on disk.
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in ggsave.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

See Also

ViolinView

Examples

```
data(mle.gene_summary)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(mle.gene_summary)
DensityView(dd, samples=c("dmso", "plx"))
#or
DensityView(dd[, c("dmso", "plx")])
```

enrich.GSE *Gene set enrichment analysis*

Description

A universal gene set enrichment analysis tools

Usage

```
enrich.GSE(geneList, keytype = "Entrez", type = "CORUM+KEGG",
  organism = "hsa", pvalueCutoff = 0.25, limit = c(1, 120),
  gmtpath = NA)
```

Arguments

geneList	A order ranked numeric vector with geneid as names.
keytype	"Entrez" or "Symbol".
type	Geneset category for testing, one of 'CORUM', 'CPX' (ComplexPortal), 'GOBP', 'GOMF', 'GOCC', 'KEGG', 'BIOCARTA', 'REACTOME', 'WikiPathways', 'EHMN', 'PID', or any combination of them (e.g. 'GOBP+GOMF+CORUM'), or 'All' (all categories).
organism	'hsa' or 'mmu'.
pvalueCutoff	Pvalue cutoff.
limit	A two-length vector (default: c(1, 120)), specifying the minimal and maximal size of gene sets for enrichent analysis.
gmtpath	The path to customized gmt file.

Value

A enrichResult instance.

Author(s)

Wubing Zhang

See Also

enrich.HGT

enrich.ORT

EnrichAnalyzer

enrichResult-class

Examples

```
data(geneList, package = "DOSE")
## Not run:
    enrichRes = enrich.GSE(geneList)
    head(slot(enrichRes, "result"))

## End(Not run)
```

enrich.HGT *Do enrichment analysis using Hypergeometric test*

Description

Do enrichment analysis using Hypergeometric test

Usage

```
enrich.HGT(geneList, keytype = "Entrez", type = "CORUM+KEGG",
  organism = "hsa", pvalueCutoff = 0.05, limit = c(1, 120),
  universe = NULL, gmtpath = NA)
```

Arguments

geneList	A numeric vector with gene as names.
keytype	"Entrez" or "Symbol".
type	Geneset category for testing, one of 'CORUM', 'CPX' (ComplexPortal), 'GOBP', 'GOMF', 'GOCC', 'KEGG', 'BIOCARTA', 'REACTOME', 'WikiPathways', 'EHMN', 'PID', or any combination of them (e.g. 'GOBP+GOMF+CORUM'), or 'All' (all categories).
organism	'hsa' or 'mmu'.
pvalueCutoff	Pvalue cutoff.
limit	A two-length vector (default: c(1, 120)), specifying the minimal and maximal size of gene sets for enrichent analysis.
universe	A character vector, specifying the backgound genelist, default is whole genome.
gmtpath	The path to customized gmt file.

Value

A enrichResult instance.

Author(s)

Wubing Zhang

See Also

enrich.GSE

enrich.ORT

EnrichAnalyzer

enrichResult-class

Examples

```
data(geneList, package = "DOSE")
genes <- geneList[1:300]
enrichRes <- enrich.HGT(genes, type = "KEGG")
head(slot(enrichRes, "result"))
```

enrich.ORT *Do enrichment analysis using over-representation test*

Description

Do enrichment analysis using over-representation test

Usage

```
enrich.ORT(geneList, keytype = "Entrez", type = "CORUM+KEGG",
  organism = "hsa", pvalueCutoff = 0.05, limit = c(1, 120),
  universe = NULL, gmtpath = NA)
```

Arguments

geneList	A numeric vector with gene as names.
keytype	"Entrez" or "Symbol".
type	Geneset category for testing, one of 'CORUM', 'CPX' (ComplexPortal), 'GOBP', 'GOMF', 'GOCC', 'KEGG', 'BIOCARTA', 'REACTOME', 'WikiPathways', 'EHMN', 'PID', or any combination of them (e.g. 'GOBP+GOMF+CORUM'), or 'All' (all categories).
organism	'hsa' or 'mmu'.
pvalueCutoff	Pvalue cutoff.
limit	A two-length vector (default: c(1, 120)), specifying the minimal and maximal size of gene sets for enrichent analysis.
universe	A character vector, specifying the backgound genelist, default is whole genome.
gmtpath	The path to customized gmt file.

Value

A enrichedResult instance.

Author(s)

Wubing Zhang

See Also

enrich.HGT

enrich.GSE

EnrichAnalyzer

enrichResult-class

Examples

```
data(geneList, package = "DOSE")
genes <- geneList[1:100]
enrichedRes <- enrich.ORT(genes)
head(slot(enrichedRes, "result"))
```

EnrichAB *Enrichment analysis for Positive and Negative selection genes*

Description

Do enrichment analysis for selected genes, in which positive selection and negative selection are termed as GroupA and GroupB

Usage

```
EnrichAB(data, pvalue = 0.25, enrich_method = "ORT",
  organism = "hsa", limit = c(1, 120), filename = NULL,
  out.dir = ".", width = 6.5, height = 4, ...)
```

Arguments

data	A data frame.
pvalue	Pvalue cutoff.
enrich_method	One of "ORT"(Over-Representing Test), "GSEA"(Gene Set Enrichment Analysis), and "HGT"(HyperGemetric test).
organism	"hsa" or "mmu".
limit	A two-length vector (default: c(1, 120)), specifying the min and max size of pathways for enrichent analysis.
filename	Suffix of output file name.
out.dir	Path to save plot to (combined with filename).
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in ggsave.

Value

A list containing enrichment results for each group genes. This list contains items four items, keggA, keggB, goA, goB. Four items are all list object, containing subitems of gridPlot and enrichRes. gridPlot is a ggplot object, and enrichRes is a enrichResult instance

Author(s)

Binbin Wang

See Also

EnrichSquare

Examples

```
data(mle.gene_summary)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(mle.gene_summary)
data=ScatterView(dd, ctrlname = "dmso", treatname = "plx")$data
## Not run:
  #GO and KEGG enrichment analysis
 enrich_result = EnrichAB(data, pvalue=0.05, organism="hsa")
  print(enrich_result$keggA$gridPlot)
  print(enrich_result$goA$gridPlot)

## End(Not run)
```

EnrichAnalyzer *Enrichment analysis*

Description

Enrichment analysis

Usage

```
EnrichAnalyzer(geneList, keytype = "Entrez", type = "CORUM+KEGG",
  method = "ORT", organism = "hsa", pvalueCutoff = 0.25,
  limit = c(1, 120), universe = NULL, filter = TRUE, gmtpath = NA)
```

Arguments

geneList	A numeric vector with gene as names.
keytype	"Entrez" or "Symbol".
type	Geneset category for testing, one of 'CORUM', 'CPX' (ComplexPortal), 'GOBP', 'GOMF', 'GOCC', 'KEGG', 'BIOCARTA', 'REACTOME', 'WikiPathways', 'EHMN', 'PID', or any combination of them (e.g. 'GOBP+GOMF+CORUM'), or 'All' (all categories).
method	One of "ORT"(Over-Representing Test), "GSEA"(Gene Set Enrichment Analysis), and "HGT"(HyperGemetric test).
organism	'hsa' or 'mmu'.
pvalueCutoff	Pvalue cutoff.
limit	A two-length vector (default: c(1, 120)), specifying the minimal and maximal size of gene sets for enrichent analysis.
universe	A character vector, specifying the backgound genelist, default is whole genome.
filter	Boolean, specifying whether filter out redundancies from the enrichment results.
gmtpath	The path to customized gmt file.

Value

enrichRes is an enrichResult instance.

Author(s)

Wubing Zhang

See Also

enrich.GSE

enrich.ORT

enrich.HGT

enrichResult-class

Examples

```
data(geneList, package = "DOSE")
keggA = EnrichAnalyzer(geneList[1:500], method = "HGT")
head(keggA@result)
```

EnrichedFilter *Simplify the enrichment results based on Jaccard index*

Description

Simplify the enrichment results based on Jaccard index

Usage

```
EnrichedFilter(enrichment = enrichment, cutoff = 0.8)
```

Arguments

enrichment	A data frame of enrichment result.
cutoff	A numeric, specifying the cutoff of Jaccard index between two pathways.

Value

A data frame.

Author(s)

Yihan Xiao

Examples

```
data(geneList, package = "DOSE")
enrichRes <- enrich.GSE(geneList)
EnrichedFilter(enrichRes)
```

EnrichedGeneView *Visualize enriched pathways and genes in those pathways*

Description

Visualize enriched pathways and genes in those pathways

Usage

```
EnrichedGeneView(enrichment, geneList, rank_by = "p.adjust", top = 5,
  bottom = 5, custom_pid = NULL, keytype = "Symbol",
  gene_cutoff = c(-log2(1.5), log2(1.5)), custom_gene = NULL,
  charLength = 40, filename = NULL, width = 7, height = 5, ...)
```

Arguments

enrichment	A data frame of enrichment result or an enrichResult object.
geneList	A numeric geneList used in enrichment anlaysis.
rank_by	"p.adjust" or "NES", specifying the indices for ranking pathways.
top	An integer, specifying the number of positively enriched terms to show.
bottom	An integer, specifying the number of negatively enriched terms to show.
custom_pid	A character vector (pathway IDs), customizing pathways to show.
keytype	"Entrez" or "Symbol".
gene_cutoff	A two-length numeric vector, specifying cutoff for genes to show.
custom_gene	A character vector (gene names), customizing genes to show.
charLength	Integer, specifying max length of enriched term name to show as coordinate lab.
filename	Figure file name to create on disk. Default filename="NULL", which means no output.
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in ggsave.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

Examples

```
data(geneList, package = "DOSE")
enrichRes <- enrich.GSE(geneList)
EnrichedGeneView(enrichment=slot(enrichRes, "result"), geneList, keytype = "Entrez")
```

EnrichedView *View enriched terms*

Description

Grid plot for enriched terms

Usage

```
EnrichedView(enrichment, rank_by = "p.adjust", top = 5, bottom = 5,
  custom_pid = NULL, x = "LogP", charLength = 40, filename = NULL,
  width = 7, height = 4, ...)
```

Arguments

enrichment	A data frame of enrichment result, with columns of ID, Description, p.adjust and NES.
rank_by	"p.adjust" or "NES", specifying the indices for ranking pathways.
top	An integer, specifying the number of top enriched terms to show.
bottom	An integer, specifying the number of bottom enriched terms to show.
custom_pid	A character vector (pathway IDs), customizing pathways to show.
x	Character, "NES" or "LogP", indicating the variable on the x-axis.
charLength	Integer, specifying max length of enriched term name to show as coordinate lab.
filename	Figure file name to create on disk. Default filename="NULL".
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in ggsave.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

See Also

EnrichedView

Examples

```
## Not run:
    data(geneList, package = "DOSE")
    enrichRes = enrich.GSE(geneList, organism="hsa")
    EnrichedView(slot(enrichRes, "result"))

## End(Not run)
```

EnrichSquare *Enrichment analysis for selected treatment related genes*

Description

Do enrichment analysis for selected treatment related genes in 9-squares

Usage

```
EnrichSquare(beta, pvalue = 0.05, enrich_method = "ORT",
  organism = "hsa", limit = c(1, 120), filename = NULL,
  out.dir = ".", width = 6.5, height = 4, ...)
```

Arguments

beta	Data frame, with rownames of Entrez IDs, which contains columns of 'group' and 'diff'.
pvalue	Pvalue cutoff.
enrich_method	One of "ORT"(Over-Representing Test) and "HGT"(HyperGemetric test).
organism	"hsa" or "mmu".
limit	A two-length vector (default: c(1, 120)), specifying the min and max size of pathways for enrichent analysis.
filename	Suffix of output file name. NULL(default) means no output.
out.dir	Path to save plot to (combined with filename).
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in ggsave.

Value

A list containing enrichment results for each group genes. This list contains several elements:

kegg1	a list record enriched KEGG pathways for Group1 genes in 9-Square
kegg2	a list record enriched KEGG pathways for Group2 genes in 9-Square
kegg3	a list record enriched KEGG pathways for Group3 genes in 9-Square
kegg4	a list record enriched KEGG pathways for Group4 genes in 9-Square
kegg12	a list record enriched KEGG pathways for Group1&Group2 genes in 9-Square
kegg13	a list record enriched KEGG pathways for Group1&Group3 genes in 9-Square
kegg24	a list record enriched KEGG pathways for Group2&Group4 genes in 9-Square
kegg34	a list record enriched KEGG pathways for Group3&Group4 genes in 9-Square
go1	a list record enriched GO terms for Group1 genes in 9-Square
go2	a list record enriched GO terms for Group2 genes in 9-Square
go3	a list record enriched GO terms for Group3 genes in 9-Square
go4	a list record enriched GO terms for Group4 genes in 9-Square
go12	a list record enriched GO terms for Group1&Group2 genes in 9-Square

go13	a list record enriched GO terms for Group1&Group3 genes in 9-Square
go24	a list record enriched GO terms for Group2&Group4 genes in 9-Square
go34	a list record enriched GO terms for Group3&Group4 genes in 9-Square

Each item in the returned list has two sub items:

gridPlot	an object created by ggplot, which can be assigned and further customized.
enrichRes	a enrichResult instance.

Author(s)

Wubing Zhang

See Also

SquareView

EnrichSquare

Examples

```
data(mle.gene_summary)
dd = ReadBeta(mle.gene_summary)
p = SquareView(dd, ctrlname = "dmso", treatname = "plx")
## Not run:
 # Read beta score from gene summary table in MAGeCK MLE results
 E1 = EnrichSquare(p$data, organism="hsa")
 print(E1$kegg1$gridPlot)

## End(Not run)
```

FluteMLE *Downstream analysis based on MAGeCK-MLE result*

Description

Integrative analysis pipeline using the gene summary table in MAGeCK MLE results

Usage

```
FluteMLE(gene_summary, ctrlname, treatname, keytype = "Symbol",
  organism = "hsa", scale_cutoff = 2, top = 10, bottom = 10,
  interestGenes = NA, limit = c(1, 120), pvalueCutoff = 0.25,
  enrich_kegg = "ORT", posControl = NULL, loess = FALSE,
  prefix = "", width = 10, height = 7, outdir = ".",
  view_allpath = FALSE)
```

Arguments

gene_summary	A data frame, which contains columns of 'Gene', ctrlname.beta and treatname.beta.
ctrlname	A character vector, specifying the names of control samples.
treatname	A character vector, specifying the names of treatment samples.
keytype	"Entrez" or "Symbol".
organism	"hsa" or "mmu".
scale_cutoff	Boolean or numeric, whether scale cutoff to whole genome level, or how many standard deviation will be used as cutoff.
top	An integer, specifying number of top selected genes to be labeled in rank figure.
bottom	An integer, specifying number of bottom selected genes to be labeled in rank figure.
interestGenes	A character vector, specifying interested genes to be labeled in rank figure.
limit	A two-length vector (default: c(1, 120)), specifying the minimal and maximal size of gene sets for enrichent analysis.
pvalueCutoff	A numeric, specifying pvalue cutoff of enrichment analysis, default 1.
enrich_kegg	One of "ORT"(Over-Representing Test), "GSEA"(Gene Set Enrichment Analysis), and "HGT"(HyperGemetric test).
posControl	A character vector, specifying a list of positive control gene symbols.
loess	Boolean, whether include loess normalization in the pipeline.
prefix	A character, indicating the prefix of output file name, which can't contain special characters.
width	The width of summary pdf in inches.
height	The height of summary pdf in inches.
outdir	Output directory on disk.
view_allpath	Boolean, whether output all pathway view figures.

Details

MAGeCK-MLE can be used to analyze screen data from multi-conditioned experiments. MAGeCK-MLE also normalizes the data across multiple samples, making them comparable to each other. The most important ouput of MAGeCK MLE is 'gene_summary' file, which includes the beta scores of multiple conditions and the associated statistics. The 'beta score' for each gene describes how the gene is selected: a positive beta score indicates a positive selection, and a negative beta score indicates a negative selection.

The downstream analysis includes identifying essential, non-essential, and target-associated genes, and performing biological functional category analysis and pathway enrichment analysis of these genes. The function also visualizes genes in the context of pathways to benefit users exploring screening data.

Value

All of the pipeline results is output into the out.dir/prefix_Results, which includes a pdf file and many folders. The pdf file 'prefix_Pipeline_results.pdf' is the summary of pipeline results. For each section in this pipeline, figures and useful data are outputed to corresponding subfolders.

- Distribution_of_BetaScores: Density plot and violin plot of beta scores.

- MAplot: Maplot for each normalized data.
- Linear_Fitting_of_BetaScores: Linear fitting of beta scores indicates the difference of cell cycle time between Control and Treatment samples.
- Scatter_Treat_Ctrl: Positive selection and negative selection.
- Enrichment_Treat-Ctrl: Enrichment analysis for positive and negative selection genes.
- Pathview_Treat_Ctrl: Pathway view for top enriched pathways.
- Scatter_9Square: Using 9 Square to select drug related genes.
- Enrichment_9Square: Enrichment analysis for selected genes.
- Pathview_9Square: Pathway view for top enriched pathways.

Author(s)

Wubing Zhang

See Also

FluteRRA

Examples

```
data(mle.gene_summary)
## Not run:
  # functional analysis for MAGeCK MLE results
  FluteMLE(mle.gene_summary, ctrlname = "dmso", treatname = "plx",
           prefix = "PLX", pvalueCutoff = 0.25, organism = "hsa")

## End(Not run)
```

FluteRRA *Downstream analysis based on MAGeCK-RRA result*

Description

Integrative analysis pipeline using the gene summary table in MAGeCK RRA results

Usage

```
FluteRRA(gene_summary, sgrna_summary, lfcCutoff = c(-1, 1),
  organism = "hsa", limit = c(1, 120), pvalueCutoff = 0.25,
  prefix = "Test", width = 12, height = 6, outdir = ".")
```

Arguments

gene_summary	A file path or a data frame of gene summary data.
sgrna_summary	A file path or a data frame of sgRNA summary data.
lfcCutoff	A two-length vector (default: c(-1, 1)), specifying the logFC cutoff for negative selection and positive selection.
organism	"hsa" or "mmu".

limit	A two-length vector (default: c(1, 120)), specifying the minimal and maximal size of gene sets for enrichent analysis.
pvalueCutoff	A numeric, specifying pvalue cutoff of enrichment analysis, default 1.
prefix	A character, indicating the prefix of output file name.
width	The width of summary pdf in inches.
height	The height of summary pdf in inches.
outdir	Output directory on disk.

Details

MAGeCK RRA allows for the comparison between two experimental conditions. It can identify genes and sgRNAs are significantly selected between the two conditions. The most important output of MAGeCK RRA is the file 'gene_summary.txt'. MAGeCK RRA will output both the negative score and positive score for each gene. A smaller score indicates higher gene importance. MAGeCK RRA will also output the statistical value for the scores of each gene. Genes that are significantly positively and negatively selected can be identified based on the p-value or FDR.

The downstream analysis of this function includes identifying positive and negative selection genes, and performing biological functional category analysis and pathway enrichment analysis of these genes.

Value

All of the pipeline results is output into the `out.dir/prefix_Results`, which includes a pdf file and a folder named 'RRA'.

Author(s)

Wubing Zhang

See Also

FluteMLE

Examples

```
data("rra.gene_summary")
data("rra.sgrna_summary")
## Not run:
    # Run the FluteRRA pipeline
    FluteRRA(rra.gene_summary, rra.sgrna_summary, prefix="RRA", organism="hsa")

## End(Not run)
```

getOrg *Determine the gene annotation package.*

Description

Determine the gene annotation package. for specific organism

Usage

```
getOrg(organism, update = FALSE)
```

Arguments

organism	Character, KEGG species code, or the common species name, used to determine the gene annotation package. For all potential values check: data(bods); bods. Default org="hsa", and can also be "human" (case insensitive).
update	Boolean, indicating whether download recent annotation from NCBI.

Value

A list containing three elements:

organism species

pkgannotation package name Symbol_Entreza data frame, mapping between gene symbol and entrez id

Author(s)

Wubing Zhang

Examples

```
ann = getOrg("human")
print(ann$pkg)
```

gsGetter *Extract pathway annotation from GMT file.*

Description

Extract pathway annotation from GMT file.

Usage

```
gsGetter(gmtpath = NA, type = "All", limit = c(0, Inf),
  organism = "hsa")
```

Arguments

gmtpath	The path to customized gmt file.
type	Geneset category for testing.
limit	A two-length vector (default: c(3, 50)), specifying the minimal and maximal size of gene sets for enrichent analysis.
organism	'hsa' or 'mmu'.

Value

A three-column data frame.

Author(s)

Wubing Zhang

Examples

```
gene2path = gsGetter()
head(gene2path)
```

HeatmapView *Draw heatmap*

Description

Draw heatmap

Usage

```
HeatmapView(mat, limit = c(-2, 2),
  colPal = rev(colorRampPalette(c("#c12603", "white", "#0073B6"), space =
  "Lab")(199)), filename = NA, width = NA, height = NA, ...)
```

Arguments

mat	Matrix like object, each row is gene and each column is sample.
limit	Max value in heatmap
colPal	colorRampPalette.
filename	File path where to save the picture.
width	Manual option for determining the output file width in inches.
height	Manual option for determining the output file height in inches.
...	Other parameters in pheatmap.

Value

Invisibly a pheatmap object that is a list with components.

Author(s)

Wubing Zhang

Examples

```
data(mle.gene_summary)
dd = ReadBeta(mle.gene_summary)
gg = cor(dd[,2:ncol(dd)])
HeatmapView(gg, display_numbers = TRUE)
```

IdentBarView *Identical bar plot*

Description

Identical bar plot

Usage

```
IdentBarView(gg, x = "x", y = "y", fill = c("#CF3C2B", "#394E80"),
  main = NULL, xlab = NULL, ylab = NULL, filename = NULL,
  width = 5, height = 4, ...)
```

Arguments

gg	A data frame.
x	A character, indicating column (in countSummary) of x-axis.
y	A character, indicating column (in countSummary) of y-axis.
fill	A character, indicating fill color of all bars.
main	A charater, specifying the figure title.
xlab	A character, specifying the title of x-axis.
ylab,	A character, specifying the title of y-axis.
filename	Figure file name to create on disk. Default filename="NULL", which means don't save the figure on disk.
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in ggsave.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

Examples

```
data(countsummary)
IdentBarView(countsummary, x="Label", y="Reads")
```

KeggPathwayView *Kegg pathway view*

Description

Plot kegg pathway and color specific genes.

Usage

```
KeggPathwayView(gene.data = NULL, cpd.data = NULL, pathway.id,
  species = "hsa", kegg.dir = ".", cpd.idtype = "kegg",
  gene.idtype = "ENTREZ", gene.annotpkg = NULL, min.nnodes = 3,
  kegg.native = TRUE, map.null = TRUE, expand.node = FALSE,
  split.group = FALSE, map.symbol = TRUE, map.cpdname = TRUE,
  node.sum = "sum", discrete = list(gene = FALSE, cpd = FALSE),
  limit = list(gene = 1, cpd = 1), bins = list(gene = 10, cpd = 10),
  both.dirs = list(gene = TRUE, cpd = TRUE), trans.fun = list(gene =
  NULL, cpd = NULL), low = list(gene = "deepskyblue1", cpd = "blue"),
  mid = list(gene = "gray", cpd = "gray"), high = list(gene = "red",
  cpd = "yellow"), na.col = "transparent", ...)
```

Arguments

gene.data	Either vector (single sample) or a matrix-like data (multiple sample). Vector should be numeric with gene IDs as names or it may also be character of gene IDs. Character vector is treated as discrete or count data. Matrix-like data structure has genes as rows and samples as columns. Row names should be gene IDs. Here gene ID is a generic concepts, including multiple types of gene, transcript and protein uniquely mappable to KEGG gene IDs. KEGG ortholog IDs are also treated as gene IDs as to handle metagenomic data. Check details for mappable ID types. Default gene.data=NULL.
cpd.data	The same as gene.data, excpet named with IDs mappable to KEGG compound IDs. Over 20 types of IDs included in CHEMBL database can be used here. Check details for mappable ID types. Default cpd.data=NULL. Note that gene.data and cpd.data can't be NULL simultaneously.
pathway.id	Character vector, the KEGG pathway ID(s), usually 5 digit, may also include the 3 letter KEGG species code.
species	Character, either the kegg code, scientific name or the common name of the target species. This applies to both pathway and gene.data or cpd.data. When KEGG ortholog pathway is considered, species="ko". Default species="hsa", it is equivalent to use either "Homo sapiens" (scientific name) or "human" (common name).
kegg.dir	Character, the directory of KEGG pathway data file (.xml) and image file (.png). Users may supply their own data files in the same format and naming convention of KEGG's (species code + pathway id, e.g. hsa04110.xml, hsa04110.png etc) in this directory. Default kegg.dir="." (current working directory).
cpd.idtype	Character, ID type used for the cpd.data. Default cpd.idtype="kegg" (include compound, glycan and drug accessions).

`gene.idtype` Character, ID type used for the gene.data, case insensitive. Default gene.idtype="entrez", i.e. Entrez Gene, which are the primary KEGG gene ID for many common model organisms. For other species, gene.idtype should be set to "KEGG" as KEGG use other types of gene IDs. For the common model organisms, you may also specify other types of valid IDs. To check the ID list, do: data(gene.idtype.list); gene.idtype.list.

`gene.annotpkg` Character, the name of the annotation package to use for mapping between other gene ID types including symbols and Entrez gene ID. Default gene.annotpkg=NULL.

`min.nnodes` Integer, minimal number of nodes of type "gene","enzyme", "compound" or "ortholog" for a pathway to be considered. Default min.nnodes=3.

`kegg.native` Logical, whether to render pathway graph as native KEGG graph (.png) or using graphviz layout engine (.pdf). Default kegg.native=TRUE.

`map.null` Logical, whether to map the NULL gene.data or cpd.data to pathway. When NULL data are mapped, the gene or compound nodes in the pathway will be rendered as actually mapped nodes, except with NA-valued color. When NULL data are not mapped, the nodes are rendered as unmapped nodes. This argument mainly affects native KEGG graph view, i.e. when kegg.native=TRUE. Default map.null=TRUE.

`expand.node` Logical, whether the multiple-gene nodes are expanded into single-gene nodes. Each expanded single-gene nodes inherits all edges from the original multiple-gene node. This option only affects graphviz graph view, i.e. when kegg.native=FALSE. This option is not effective for most metabolic pathways where it conflits with converting reactions to edges. Default expand.node=FLASE.

`split.group` Logical, whether split node groups are split to individual nodes. Each split member nodes inherits all edges from the node group. This option only affects graphviz graph view, i.e. when kegg.native=FALSE. This option also effects most metabolic pathways even without group nodes defined orginally. For these pathways, genes involved in the same reaction are grouped automatically when converting reactions to edges unless split.group=TRUE. d split.group=FLASE.

`map.symbol` Logical, whether map gene IDs to symbols for gene node labels or use the graphic name from the KGML file. This option is only effective for kegg.native=FALSE or same.layer=FALSE when kegg.native=TRUE. For same.layer=TRUE when kegg.native=TRUE, the native KEGG labels will be kept. Default map.symbol=TRUE.

`map.cpdname` Logical, whether map compound IDs to formal names for compound node labels or use the graphic name from the KGML file (KEGG compound accessions). This option is only effective for kegg.native=FALSE. When kegg.native=TRUE, the native KEGG labels will be kept. Default map.cpdname=TRUE.

`node.sum` Character, the method name to calculate node summary given that multiple genes or compounds are mapped to it. Poential options include "sum","mean", "median", "max", "max.abs" and "random". Default node.sum="sum".

`discrete` A list of two logical elements with "gene" and "cpd" as the names. This argument tells whether gene.data or cpd.data should be treated as discrete. Default dsicrete=list(gene=FALSE, cpd=FALSE), i.e. both data should be treated as continuous.

`limit` A list of two numeric elements with "gene" and "cpd" as the names. This argument specifies the limit values for gene.data and cpd.data when converting them to pseudo colors. Each element of the list could be of length 1 or 2. Length 1 suggests discrete data or 1 directional (positive-valued) data, or the absolute limit for 2 directional data. Length 2 suggests 2 directional data. Default limit=list(gene=1, cpd=1).

bins	A list of two integer elements with "gene" and "cpd" as the names. This argument specifies the number of levels or bins for gene.data and cpd.data when converting them to pseudo colors. Default limit=list(gene=10, cpd=10).
both.dirs	A list of two logical elements with "gene" and "cpd" as the names. This argument specifies whether gene.data and cpd.data are 1 directional or 2 directional data when converting them to pseudo colors. Default limit=list(gene=TRUE, cpd=TRUE).
trans.fun	A list of two function (not character) elements with "gene" and "cpd" as the names. This argument specifies whether and how gene.data and cpd.data are transformed. Examples are log, abs or users' own functions. Default limit=list(gene=NULL, cpd=NULL).
low	A list of two colors with "gene" and "cpd" as the names.
mid	A list of two colors with "gene" and "cpd" as the names.
high	A list of two colors with "gene" and "cpd" as the names.
na.col	Color used for NA's or missing values in gene.data and cpd.data. d na.col="transparent".
...	Extra arguments passed to keggview.native or keggview.graph function.

Details

The function KeggPathwayView is a revised version of pathview function in pathview package. KeggPathwayView maps and renders user data on relevant pathway graphs. KeggPathwayView is a stand alone program for pathway based data integration and visualization. It also seamlessly integrates with pathway and functional analysis tools for large-scale and fully automated analysis. KeggPathwayView provides strong support for data Integration. It works with: 1) essentially all types of biological data mappable to pathways, 2) over 10 types of gene or protein IDs, and 20 types of compound or metabolite IDs, 3) pathways for over 2000 species as well as KEGG orthology, 4) varoius data attributes and formats, i.e. continuous/discrete data, matrices/vectors, single/multiple samples etc. To see mappable external gene/protein IDs do: data(gene.idtype.list), to see mappable external compound related IDs do: data(rn.list); names(rn.list). KeggPathwayView generates both native KEGG view and Graphviz views for pathways. Currently only KEGG pathways are implemented. Hopefully, pathways from Reactome, NCI and other databases will be supported in the future.

The argument `low`, `mid`, and `high` specifies the color spectra to code gene.data and cpd.data. When data are 1 directional (TRUE value in both.dirs), only mid and high are used to specify the color spectra. Default spectra (low-mid-high) "green"-"gray"-"red" and "blue"-"gray"-"yellow" are used for gene.data and cpd.data respectively. The values for 'low, mid, high' can be given as color names ('red'), plot color index (2=red), and HTML-style RGB, ("\#FF0000"=red).

Value

The result returned by KeggPathwayView function is a named list corresponding to the input pathway ids. Each element (for each pathway itself is a named list, with 2 elements ("plot.data.gene", "plot.data.cpd"). Both elements are data.frame or NULL depends on the corresponding input data gene.data and cpd.data. These data.frames record the plot data for mapped gene or compound nodes: rows are mapped genes/compounds, columns are:

kegg.names	standard KEGG IDs/Names for mapped nodes. It's Entrez Gene ID or KEGG Compound Accessions.
labels	Node labels to be used when needed.
all.mapped	All molecule (gene or compound) IDs mapped to this node.

type	node type, currently 4 types are supported: "gene","enzyme", "compound" and "ortholog".
x	x coordinate in the original KEGG pathway graph.
y	y coordinate in the original KEGG pathway graph.
width	node width in the original KEGG pathway graph.
height	node height in the original KEGG pathway graph.
other columns	columns of the mapped gene/compound data and corresponding pseudo-color codes for individual samples

Author(s)

Wubing Zhang

See Also

pathview

Examples

```
#load data
data(mle.gene_summary)
dd = ReadBeta(mle.gene_summary)
gene.data = dd$plx
names(gene.data) = rownames(dd)
## Not run:
pv.out <- KeggPathwayView(gene.data, pathway.id = "04110",
  species = "hsa", out.suffix = "gse16873", kegg.native = TRUE)

## End(Not run)
```

MapRatesView *View mapping ratio*

Description

View mapping ratio of each sample

Usage

```
MapRatesView(countSummary, Label = "Label", Reads = "Reads",
  Mapped = "Mapped", filename = NULL, width = 5, height = 4, ...)
```

Arguments

countSummary	A data frame, which contains columns of 'Label', 'Reads', and 'Mapped'
Label	A character, indicating column (in countSummary) of sample names.
Reads	A character, indicating column (in countSummary) of total reads.
Mapped	A character, indicating column (in countSummary) of mapped reads.
filename	Figure file name to create on disk. Default filename="NULL", which means don't save the figure on disk.

width	As in ggsave.
height	As in ggsave.
...	Other available parameters in ggsave.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

Examples

```
data(countsummary)
MapRatesView(countsummary)
```

MAView *MAplot of gene beta scores*

Description

MAplot of gene beta scores in Control vs Treatment

Usage

```
MAView(beta, ctrlname = "Control", treatname = "Treatment",
  main = NULL, show.statistics = TRUE, add.smooth = TRUE, lty = 1,
  smooth.col = "red", plot.method = c("loess", "lm", "glm", "gam"),
  filename = NULL, width = 5, height = 4, ...)
```

Arguments

beta	Data frame, including ctrlname and treatname as columns.
ctrlname	Character vector, specifying the name of control sample.
treatname	Character vector, specifying the name of treatment sample.
main	As in plot.
show.statistics	Show statistics .
add.smooth	Whether add a smooth line to the plot.
lty	Line type for smooth line.
smooth.col	Color of smooth line.
plot.method	A string specifying the method to fit smooth line, which should be one of "loess" (default), "lm", "glm" and "gam".
filename	Figure file name to create on disk. Default filename="NULL", which means don't save the figure on disk.
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in function 'ggsave'.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

Examples

```
data(mle.gene_summary)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(mle.gene_summary)
MAView(dd, ctrlname = "dmso", treatname = "plx")
```

---

mle.gene_summary *Gene summary table in MAGeCK MLE results*

Description

The gene summary results generated by running MAGeCK MLE on CRISPR screens.

Usage

```
data("mle.gene_summary")
```

Format

A data frame.

References

https://www.ncbi.nlm.nih.gov/pubmed/25494202 https://www.ncbi.nlm.nih.gov/pubmed/26673418

Examples

```
data("mle.gene_summary")
head(mle.gene_summary)
```

noEnrichPlot *Blank figure*

Description

Blank figure

Usage

```
noEnrichPlot(main = "No enriched terms")
```

Arguments

main	The title of figure.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

normalize.loess *normalize.loess*

Description

Loess normalization method.

Usage

```
normalize.loess(mat, subset = sample(1:(dim(mat)[1]), min(c(5000,
  nrow(mat)))), epsilon = 10^-2, maxit = 1, log.it = FALSE,
  verbose = TRUE, span = 2/3, family.loess = "symmetric", ...)
```

Arguments

mat	A matrix with columns containing the values of the chips to normalize.
subset	A subset of the data to fit a loess to.
epsilon	A tolerance value (supposed to be a small value - used as a stopping criterion).
maxit	Maximum number of iterations.
log.it	Logical. If TRUE it takes the log2 of mat.
verbose	Logical. If TRUE displays current pair of chip being worked on.
span	Parameter to be passed the function loess
family.loess	Parameter to be passed the function loess. "gaussian" or "symmetric" are acceptable values for this parameter.
...	Any of the options of normalize.loess you would like to modify (described above).

Value

A matrix similar as mat.

Author(s)

Wubing Zhang

See Also

loess

NormalizeBeta

Examples

```
beta = ReadBeta(mle.gene_summary)
beta_loess = normalize.loess(beta[,c("dmso", "plx")])
```

NormalizeBeta *Normalize gene beta scores*

Description

Two normalization methods are available. cell_cycle method normalizes gene beta scores based on positive control genes in CRISPR screening. loess method normalizes gene beta scores using loess.

Usage

```
NormalizeBeta(beta, id = 1, method = "cell_cycle", posControl = NULL,
  samples = NULL)
```

Arguments

beta	Data frame.
id	An integer specifying the column of gene.
method	Character, one of 'cell_cycle'(default) and 'loess'. or character string giving the name of the table column containing the gene names.
posControl	A character vector, specifying a list of positive control genes.
samples	Character vector, specifying the sample names in *beta* columns. If NULL (default), take all *beta* columns as samples.

Details

In CRISPR screens, cells treated with different conditions (e.g., with or without drug) may have different proliferation rates. So it's necessary to normalize the proliferation rate based on defined positive control genes among samples. After normalization, the beta scores are comparable across samples. loess is another optional normalization method, which is used to normalize array data before.

Value

A data frame with same format as input data *beta*.

Author(s)

Wubing Zhang

Examples

```
data(mle.gene_summary)
data(Zuber_Essential)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(mle.gene_summary)
#Cell Cycle normalization
dd_essential = NormalizeBeta(dd, samples=c("dmso", "plx"),
    method="cell_cycle", posControl = Zuber_Essential$GeneSymbol)
head(dd_essential)

#Optional loess normalization
dd_loess = NormalizeBeta(dd, samples=c("dmso", "plx"), method="loess")
head(dd_loess)
```

`RankView` *View the rank of gene points*

Description

Rank all genes according to beta score deviation, and label top and bottom meaningful genes. Some other interested genes can be labeled too.

Usage

```
RankView(rankdata, genelist = NULL, top = 10, bottom = 10,
  cutoff = NULL, main = NULL, filename = NULL, width = 5,
  height = 4, ...)
```

Arguments

`rankdata`	Numeric vector, with gene as names.
`genelist`	Character vector, specifying genes to be labeled in figure.
`top`	Integer, specifying number of top genes to be labeled.
`bottom`	Integer, specifying number of bottom genes to be labeled.
`cutoff`	Numeric.
`main`	As in 'plot'.
`filename`	Figure file name to create on disk. Default filename="NULL", which means no output.
`width`	As in ggsave.
`height`	As in ggsave.
`...`	Other available parameters in function 'ggsave'.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

Examples

```
data(rra.gene_summary)
rra = ReadRRA(rra.gene_summary)
rankdata = rra$LFC
names(rankdata) = rra$Official
RankView(rankdata)
```

ReadBeta *Read gene beta scores*

Description

Read gene beta scores from file or data frame

Usage

```
ReadBeta(gene_summary)
```

Arguments

gene_summary	A file path or a data frame, which has columns of 'Gene' and beta score of samples.

Value

A data frame, in which the first column is ENTREZID, and the later columns are beta score for each samples.

Author(s)

Wubing Zhang

Examples

```
data(mle.gene_summary)
dd = ReadBeta(mle.gene_summary)
head(dd)
```

ReadGMT *ReadGMT*

Description

Parse gmt file to a data.frame

Usage

```
ReadGMT(gmtpath, limit = c(0, Inf))
```

Arguments

gmtpath	The path to gmt file.
limit	A integer vector of length two, specifying the limit of geneset size.

Value

An data.frame, in which the first column is gene, and the second column is pathway name.

Author(s)

Wubing Zhang

ReadRRA *Read gene summary file in MAGeCK-RRA results*

Description

Read gene summary file in MAGeCK-RRA results

Usage

```
ReadRRA(gene_summary)
```

Arguments

gene_summary	A file path or a data frame of gene summary data generated by command ‘mageck test‘.

Value

A data frame including four columns, named "Official", "EntrezID", "LFC" and "FDR".

Author(s)

Wubing Zhang

Examples

```
data(rra.gene_summary)
dd.rra = ReadRRA(rra.gene_summary)
head(dd.rra)
```

ReadsgRRA *Read sgRNA summary in MAGeCK-RRA results*

Description

Read sgRNA summary in MAGeCK-RRA results

Usage

```
ReadsgRRA(sgRNA_summary)
```

Arguments

sgRNA_summary	A file path or a data frame of sgRNA summary data.

Value

A data frame.

Author(s)

Wubing Zhang

Examples

```
data(rra.sgrna_summary)
sgrra = ReadsgRRA(rra.sgrna_summary)
head(sgrra)
```

---

rra.gene_summary *Gene summary data generated by running MAGeCK RRA*

Description

The gene summary results generated by running MAGeCK on CRISPR screens.

Usage

```
data("rra.gene_summary")
```

Format

A data frame.

References

https://www.ncbi.nlm.nih.gov/pubmed/25494202 https://www.ncbi.nlm.nih.gov/pubmed/25476604

Examples

```
data("rra.gene_summary")
head(rra.gene_summary)
```

`rra.sgrna_summary` *sgRNA summary data generated by running MAGeCK RRA*

Description

The sgRNA summary results generated by running ‘mageck test‘ on CRISPR screens.

Usage

```
data("rra.sgrna_summary")
```

Format

A data frame.

References

https://www.ncbi.nlm.nih.gov/pubmed/25494202 https://www.ncbi.nlm.nih.gov/pubmed/25476604

Examples

```
data(rra.sgrna_summary)
head(rra.sgrna_summary)
```

`ScatterView` *Scatter plot*

Description

Scatter plot of all genes, in which x-axis is mean beta score in Control samples, y-axis is mean beta scores in Treatment samples.

Usage

```
ScatterView(beta, ctrlname = "Control", treatname = "Treatment",
  scale_cutoff = 2, main = NULL, filename = NULL, width = 5,
  height = 4, ...)
```

Arguments

beta	Data frame, including ctrlname and treatname as columns.
ctrlname	A character, specifying the names of control samples.
treatname	A character, specifying the names of treatment samples.
scale_cutoff	Boolean or numeric, whether scale cutoff to whole genome level, or how many standard deviation will be used as cutoff.
main	As in 'plot'.
filename	Figure file name to create on disk. Default filename="NULL", which means don't save the figure on disk.
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in function 'ggsave'.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

See Also

SquareView

Examples

```
data(mle.gene_summary)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(mle.gene_summary)
ScatterView(dd, ctrlname = "dmso", treatname = "plx")
```

Selector *Select signatures from candidate list (according to the consistence in most samples).*

Description

Select signatures from candidate list (according to the consistence in most samples).

Usage

```
Selector(mat, cutoff = 0, type = "<", select = 0.8)
```

Arguments

mat	Data matrix, each row is candidates (genes), each column is samples.
cutoff	Cutoff to define the signatures.
type	Direction to select signatures.
select	Proportion of samples in which signature is selected.

Value

An list containing two elements, first is selected signature and second is a ggplot object.

Examples

```
mat = matrix(rnorm(1000*30), 1000, 30)
rownames(mat) = paste0("Gene", 1:1000)
colnames(mat) = paste0("Sample", 1:30)
hits = Selector(mat, select = 0.68)
print(hits$p)
```

---

sgRankView *View sgRNA rank.*

Description

View sgRNA rank.

Usage

```
sgRankView(df, gene = NULL, top = 3, bottom = 3, neg_ctrl = NULL,
  binwidth = 0.3, interval = 0.1, bg.col = "gray90",
  filename = NULL, width = 5, height = 3.5, ...)
```

Arguments

df	A data frame, which contains columns of 'sgrna', 'Gene', and 'LFC'.
gene	Character vector, specifying genes to be plotted.
top	Integer, specifying number of top genes to be plotted.
bottom	Integer, specifying number of bottom genes to be plotted.
neg_ctrl	A vector specifying negative ctrl genes.
binwidth	A numeric value specifying the bar width.
interval	A numeric value specifying the interval length between each bar.
bg.col	A character value specifying the background color.
filename	Figure file name to create on disk. Default filename="NULL", which means no output.
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in function 'ggsave'.

Value

An object created by ggplot.

Author(s)

Yihan Xiao

Examples

```
data(rra.sgrna_summary)
sgrra = ReadsgRRA(rra.sgrna_summary)
sgRankView(sgrra)
```

SquareView *Scatter plot of 9-Square*

Description

Plot a scatter plot with Control beta score as x-axis and Treatment beta score as y-axis, and colored treatment related genes.

Usage

```
SquareView(beta, ctrlname = "Control", treatname = "Treatment",
  label = 0, label.top = TRUE, top = 5, genelist = c(),
  x_cutoff = NULL, y_cutoff = NULL, intercept = NULL,
  groups = c("midleft", "topcenter", "midright", "bottomcenter"),
  groupnames = paste0("Group", 1:length(groups)), main = NULL,
  filename = NULL, width = 6, height = 4, ...)
```

Arguments

beta	Data frame, including columns of *ctrlname* and *treatname*, with Gene Symbol as rowname.
ctrlname	A character, specifying the names of control samples.
treatname	A character, specifying the name of treatment samples.
label	An integer or a character specifying the column used as the label, default value is 0 (row names).
label.top	Boolean, whether label the top selected genes, default label the top 10 genes in each group.
top	Integer, specifying the number of top selected genes to be labeled. Default is 5.
genelist	Character vector, specifying labeled genes.
x_cutoff	An one or two-length numeric vector, specifying the cutoff used for x-axis.
y_cutoff	An one or two-length numeric vector, specifying the cutoff used for y-axis.
intercept	An one or two-length numeric vector, specifying the intercept of diagonal.
groups	A character vector, specifying which group to be colored. Optional groups include "topleft", "topcenter", "topright", "midleft", "midright", "bottomleft", "bottomcenter", "bottomright".

groupnames	A character vector, specifying group names.
main	As in 'plot'.
filename	Figure file name to create on disk. Default filename="NULL", which means don't save the figure on disk.
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in function 'ggsave'.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

See Also

ScatterView

Examples

```
data(mle.gene_summary)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(mle.gene_summary)
SquareView(dd, ctrlname = "dmso", treatname = "plx", label = "Gene")
```

TransGeneID *Gene ID conversion between ENTREZID and SYMBOL*

Description

Gene ID conversion between ENTREZID and SYMBOL

Usage

```
TransGeneID(genes, fromType = "Symbol", toType = "Entrez",
  organism = "hsa", useBiomart = FALSE,
  ensemblHost = "www.ensembl.org")
```

Arguments

genes	A character vector, input genes to be converted.
fromType	The input ID type, one of "Symbol" (default), "Entrez" and "Ensembl"; you can also input other valid attribute names for biomaRt.
toType	The output ID type, one of "Symbol", "Entrez" (default), "Ensembl"; you can also input other valid attribute names for biomaRt.

organism	One of "hsa"(or 'Human'), "mmu"(or 'Mouse'), "bta", "cfa", "ptr", "rno", and "ssc".
useBiomart	Boolean, indicating whether use Biomart to do the transformation.
ensemblHost	String, specifying ensembl host, you can use 'listEnsemblArchives()' to show all available Ensembl archives hosts.

Value

A character vector, named by unique input gene ids.

Author(s)

Wubing Zhang

See Also

eg2id

Examples

```
data(mle.gene_summary)
TransGeneID(mle.gene_summary$Gene[1:10], organism="hsa")
```

ViolinView *Violin plot*

Description

Plots the violin of beta scores in Control and Treatment samples.

Usage

```
ViolinView(beta, samples = NULL, main = NULL, ylab = "Beta Score",
  filename = NULL, width = 5, height = 4, ...)
```

Arguments

beta	Data frame, , including samples as columns.
samples	Character, specifying the name of samples to be compared.
main	As in 'plot'.
ylab	As in 'plot'.
filename	Figure file name to create on disk. Default filename="NULL", which means don't save the figure on disk.
width	As in ggsave.
height	As in ggsave.
...	Other available parameters in function 'ggsave'.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

See Also

DensityView

Examples

```
data(mle.gene_summary)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(mle.gene_summary)
ViolinView(dd, samples=c("dmso", "plx"))
#or
ViolinView(dd[, c("dmso", "plx")])
```

VolcanoView *Volcano View*

Description

Volcano plot

Usage

```
VolcanoView(df, x = "logFC", y = "adj.P.Val", Label = NA, top = 5,
  topnames = NULL, x_cutoff = log2(1.5), y_cutoff = 0.05,
  mycolour = c("gray80", "#e41a1c", "#377eb8"), alpha = 0.6,
  main = NULL, xlab = "Log2 Fold Change", ylab = "-Log10(Adjust.P)",
  filename = NULL, width = 4, height = 2.5, ...)
```

Arguments

df	Data frame
x	Colname of df specifying x-axis in Volcanno figure, 'logFC' (default).
y	Colname of df specifying y-axis in Volcanno figure, 'adj.P.Val' (default), which will be plot after log10 transformation.
Label	Colname of df specifying labeled terms in Volcanno figure.
top	Interger, the number of top significant terms to be labeled.
topnames	Character vector, indicating interested terms to be labeled.
x_cutoff	Cutoff of x-axis.
y_cutoff	Cutoff of y-axis.
mycolour	A color vector, specifying colors of non-significant, significant up and down-regulated genes.

alpha	Parameter in ggplot.
main	Title of volcano figure.
xlab	Label of x-axis in figure.
ylab	Label of y-axis in figure.
filename	Figure file name to create on disk. Default filename="NULL", which means don't save the figure on disk.
width	Width of figure.
height	Height of figure.
...	Other available parameters in ggsave.

Value

An object created by ggplot, which can be assigned and further customized.

Author(s)

Wubing Zhang

Examples

```
data(rra.gene_summary)
rra = ReadRRA(rra.gene_summary)
VolcanoView(rra, x = "LFC", y = "FDR", Label = "Official")
```

Zuber_Essential *Core essential gene list*

Description

A gene list of core essential genes

Usage

```
data("Zuber_Essential")
```

Format

A dataframe including 664 rows, representing 664 core essential gene.

Index