Package ‘MAGeCKFlute’

October 16, 2018

Type Package

Title Integrative analysis pipeline for pooled CRISPR functional genetic screens

Version 1.0.1

Date 2018-10-05

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, pathview, gridExtra

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

Imports ggExtra, ggsci, ggrepel, clusterProfiler, png, data.table, pheatmap, RColorBrewer, sva, GOstats, Category, DOSE, biomaRt, grid

LazyData TRUE

NeedsCompilation no

biocViews Workflow, CRISPR, PooledScreens, QualityControl, Normalization, MultipleComparison, FunctionalGenomics, GeneSetEnrichment, Pathways, Visualization

RoxygenNote 6.0.1

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

git_branch RELEASE_3_7

git_last_commit 87b8287

git_last_commit_date 2018-10-05

Date/Publication 2018-10-15

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(), organism = "hsa",
  view_allpath = FALSE, title = "Group A",
  sub = "Negative control normalized", 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.

`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 4 enriched pathways.

`title` optional string, or grob.

`sub` optional string, or grob.

`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

```
## Not run:
  data(MLE_Data)
  # Read beta score from gene summary table in MAGeCK MLE results
  dd = ReadBeta(MLE_Data, organism="hsa")
  tmp = TransGeneID(rownames(dd), "Symbol", "Entrez", organism = "hsa")
  idx = is.na(tmp) | duplicated(tmp)
  dd = dd[!idx,]
  rownames(dd) = tmp[!idx]
```

```
    dd$Control = rowMeans(dd[, 1:2])
    dd$Treatment = rowMeans(dd[, 3:4])
    arrangePathview(dd, "hsa00534", title=NULL, sub=NULL, organism="hsa")

## End(Not run)
```

BatchRemove *Batch effect removal*

Description

Remove batch effect

Usage

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

Arguments

mat	Matrix, or a file path of data.
batchMat	Matrix like data object or a file path of batch table, which has at least two columns, including Samples(matched colname of mat) and Batch. It can have the third column, which should be Covariate.
log2trans	Boolean, specifying whether do log2 transition before batch removal.
positive	Boolean, specifying whether all values should be positive.

Value

A list contrains two objects, including data and p.

Author(s)

Wubing Zhang

See Also

ComBat

Examples

```
data(MLE_Data)
beta = ReadBeta(MLE_Data, organism="hsa")
samples = c("D7_R1", "D7_R2", "PLX7_R1", "PLX7_R2")
batchMat = data.frame(samples = samples, batch = c("bat1","bat2","bat1","bat2"), cov = c(1,1,2,2))
res = BatchRemove(beta[, samples], batchMat)
```

CellCycleView — *Estimate cell cycle time for all samples compared to control sample and view.*

Description

Estimate cell cycle time in different samples by linear fitting of beta scores, and plot fitting lines, in which x-axis is control beta score and y-axis is beta score of all samples.

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_Data)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(MLE_Data, organism="hsa")
CellCycleView(dd, ctrlname = c("D7_R1", "D7_R2"), treatname = c("PLX7_R1", "PLX7_R2"))
```

CorrView *Visualize the correlation between two object*

Description

Visualize the correlation between two object

Usage

```
CorrView(gg, x, y, smoothMethod = "lm", 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.
`smoothMethod`	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.

Examples

```
gg = data.frame(x = rnorm(50), y = rnorm(50))
CorrView(gg, x="x", y="y")
```

`CutoffCalling` *Call cutoff*

Description

Calculate standard deviation as cutoff for a numeric vector

Usage

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

Arguments

`d`	A numeric vector.
`scale`	Boolean or numeric, whether scale cutoff to whole genome level, or how many standard deviation will be used as cutoff.

Value

A numeric value.

`DensityDiffView` *Density plot for beta score deviation between Control and Treatment*

Description

Plot the density of beta score deviation between two samples.

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_Data)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(MLE_Data, organism="hsa")
# Density plot of beta score deviation between control and treatment
DensityDiffView(dd, ctrlname = "D7_R1", treatname = "PLX7_R1")
```

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_Data)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(MLE_Data, organism="hsa")
DensityView(dd, samples=c("D7_R1", "D7_R2", "PLX7_R1", "PLX7_R2"))
#or
DensityView(dd[, c("D7_R1", "D7_R2", "PLX7_R1", "PLX7_R2")])
```

enrich.DAVID *Do enrichment analysis using DAVID*

Description

an update version of DAVIDWebService to do enrichment analysis

Usage

```
enrich.DAVID(gene, universe = NULL, david.user, idType = "ENTREZ_GENE_ID",
  minGSSize = 2, maxGSSize = 500, annotation = "GOTERM_BP_FAT",
  pvalueCutoff = 0.25, pAdjustMethod = "BH", qvalueCutoff = 0.2)
```

Arguments

gene	Character vector, specifying the genelist to do enrichment analysis.
universe	Character vector, specifying the backgound genelist, default is whole genome.
david.user	Character, specifying a valid DAVID user account.
idType	Character, indicating the gene id type of input genelist, such as "ENTREZ_GENE_ID"(default).
minGSSize	Minimal size of each geneSet for testing.
maxGSSize	Maximal size of each geneSet for analyzing.
annotation	Geneset category for testing, GOTERM_BP_FAT(default).
pvalueCutoff	Pvalue cutoff.
pAdjustMethod	One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
qvalueCutoff	Qvalue cutoff.

Value

A enrichResult instance.

Author(s)

Wubing Zhang

See Also

enrich.HGT

enrich.GOstats

enrich.GSE

enrich.ORT

enrichment_analysis

enrichResult-class

Examples

```
data(geneList, package = "DOSE")
genes <- names(geneList)[1:100]
## Not run:
# Before running this example, you need to have a david account.
enrichRes <- enrich.DAVID(genes, david.user="david.user@edu.com")
head(enrichRes@result)

## End(Not run)
```

`enrich.GOstats` *Do enrichment analysis using GOstats*

Description

Do enrichment analysis using GOstats method

Usage

```
enrich.GOstats(gene, universe = NULL, type = c("KEGG", "BP", "MF", "CC"),
  organism = "hsa", pvalueCutoff = 0.25, pAdjustMethod = "BH")
```

Arguments

`gene`	A character vector, specifying the genelist to do enrichment analysis.
`universe`	A character vector, specifying the backgound genelist, default is whole genome.
`type`	Geneset category for testing, KEGG(default).
`organism`	A character, specifying organism, such as "hsa" or "Human"(default), and "mmu" or "Mouse"
`pvalueCutoff`	Pvalue cutoff.
`pAdjustMethod`	One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

Value

A enrichResult instance.

Author(s)

Wubing Zhang

See Also

`enrich.HGT`

`enrich.DAVID`

`enrich.GSE`

`enrich.ORT`

`enrichment_analysis`

`enrichResult-class`

Examples

```
data(geneList, package = "DOSE")
genes <- names(geneList)[1:100]
## Not run:
    enrichRes <- enrich.GOstats(genes, type="BP")
    head(enrichRes@result)

## End(Not run)
```

`enrich.GSE` *GSEA*

Description

A universal gene set enrichment analysis tools

Usage

```
enrich.GSE(geneList, type = "MsigDB_c2_h", organism = "hsa",
  minGSSize = 10, maxGSSize = 500, pvalueCutoff = 0.25,
  pAdjustMethod = "BH")
```

Arguments

`geneList`	A order ranked numeric vector with geneid as names.
`type`	A character, indicating geneset category for testing, "MsigDB_c2_h"(default).
`organism`	A character, specifying organism, only 'human' is available.
`minGSSize`	Minimal size of each geneSet for testing.
`maxGSSize`	Maximal size of each geneSet for analyzing.
`pvalueCutoff`	Pvalue cutoff.
`pAdjustMethod`	One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

Value

A enrichResult instance.

Author(s)

Wubing Zhang

See Also

`enrich.HGT`

`enrich.DAVID`

`enrich.GOstats`

`enrich.ORT`

`enrichment_analysis`

`enrichResult-class`

Examples

```
data(geneList, package = "DOSE")
## Not run:
    enrichRes = enrich.GSE(geneList, type = "KEGG", organism="hsa")
    head(enrichRes@result)

## End(Not run)
```

---

enrich.HGT *Do enrichment analysis using Hypergeometric test*

Description

Do enrichment analysis using Hypergeometric test

Usage

```
enrich.HGT(gene, universe = NULL, type = "KEGG", organism = "hsa",
  pvalueCutoff = 0.25, pAdjustMethod = "BH", minGSSize = 2,
  maxGSSize = 500)
```

Arguments

gene	A character vector, specifying the genelist to do enrichment analysis.
universe	A character vector, specifying the backgound genelist, default is whole genome.
type	Geneset category for testing, KEGG(default).
organism	A character, specifying organism, such as "hsa" or "Human"(default), and "mmu" or "Mouse"
pvalueCutoff	Pvalue cutoff.
pAdjustMethod	One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
minGSSize	Minimal size of each geneSet for testing.
maxGSSize	Maximal size of each geneSet for analyzing.

Value

A enrichResult instance.

Author(s)

Feizhen Wu

See Also

enrich.GOstats

enrich.DAVID

enrich.GSE

enrich.ORT

enrichment_analysis

enrichResult-class

Examples

```
data(geneList, package = "DOSE")
genes <- names(geneList)[1:100]
enrichRes <- enrich.HGT(genes)
head(enrichRes@result)
```

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

Description

Do enrichment analysis using over-representation test

Usage

```
enrich.ORT(gene, universe = NULL, type = "KEGG", organism = "hsa",
  pvalueCutoff = 0.25, qvalueCutoff = 0.2, pAdjustMethod = "BH",
  minGSSize = 2, maxGSSize = 50)
```

Arguments

`gene`	A character vector, specifying the genelist to do enrichment analysis.
`universe`	A character vector, specifying the backgound genelist, default is whole genome.
`type`	Geneset category for testing, KEGG(default).
`organism`	A character, specifying organism, such as "hsa" or "Human"(default), and "mmu" or "Mouse"
`pvalueCutoff`	Pvalue cutoff.
`qvalueCutoff`	Qvalue cutoff.
`pAdjustMethod`	One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
`minGSSize`	Minimal size of each geneSet for testing.
`maxGSSize`	Maximal size of each geneSet for analyzing.

Value

A enrichResult instance.

Author(s)

Wubing Zhang

See Also

`enrich.HGT`

`enrich.DAVID`

`enrich.GOstats`

`enrich.GSE`

`enrichment_analysis`

`enrichResult-class`

Examples

```
data(geneList, package = "DOSE")
genes <- names(geneList)[1:100]
enrichRes <- enrich.ORT(genes)
head(enrichRes@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",
  adjust = "BH", filename = NULL, out.dir = ".", gsea = FALSE,
  width = 6.5, height = 4, ...)
```

Arguments

`data`	A data frame containing columns "diff", with rownames of Entrez IDs.
`pvalue`	Pvalue cutoff.
`enrich_method`	One of "ORT"(Over-Representing Test), "DAVID", "GOstats", and "HGT"(HyperGemetric test).
`organism`	A character, specifying organism, such as "hsa" or "Human"(default), and "mmu" or "Mouse"
`adjust`	One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
`filename`	Suffix of output file name. NULL(default) means no output.
`out.dir`	Path to save plot to (combined with filename).
`gsea`	Boolean, specifying if do GSEA for GroupA and GroupB genes. Default gsea=FALSE.
`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`, `bpA`, `bpB`. 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`

data(MLE_Data) # Read beta score from gene summary table in MAGeCK MLE results

EnrichedGSEView *View enriched terms in GSEA*

Description

Grid plot for enriched terms in GSEA

Usage

```
EnrichedGSEView(enrichment, plotTitle = NULL, termNum = 15,
  charLength = 40, filename = NULL, width = 5, height = 4, ...)
```

Arguments

`enrichment`	A data frame of enrichment result, with columns of ID, Description, p.adjust and NES
`plotTitle`	Same as 'title' in 'plot'.
`termNum`	Integer, specifying number of top enriched terms 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

See Also

`EnrichedView`

Examples

```
## Not run:
    data(geneList, package = "DOSE")
    enrichRes = enrich.GSE(geneList, type = "KEGG", organism="hsa")
    EnrichedGSEView(enrichRes@result, plotTitle = "GSEA Analysis")

## End(Not run)
```

EnrichedView *View enriched terms*

Description

Grid plot for enriched terms

Usage

```
EnrichedView(enrichment, plotTitle = NULL, color = "#3f90f7",
  termNum = 15, charLength = 40, filename = NULL, width = 5,
  height = 4, ...)
```

Arguments

`enrichment`	A data frame of enrichment result, with columns of ID, Description, p.adjust and Count.
`plotTitle`	Same as 'title' in 'plot'.
`color`	Color of nodes.
`termNum`	Integer, specifying number of top enriched terms 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)

Feizhen Wu

See Also

`KeggPathwayView`

`EnrichedGSEView`

Examples

```
data(geneList, package = "DOSE")
genes <- names(geneList)[1:100]
enrichRes <- enrich.HGT(genes)
EnrichedView(enrichment=enrichRes@result)
```

enrichment_analysis *Enrichment analysis*

Description

Enrichment analysis

Usage

```
enrichment_analysis(geneList, universe = NULL, method = "ORT",
  type = "KEGG", organism = "hsa", pvalueCutoff = 0.25,
  qvalueCutoff = 0.2, pAdjustMethod = "BH", minGSSize = 2,
  maxGSSize = 50, plotTitle = NULL, color = "#3f90f7")
```

Arguments

`geneList`	A character vector or a ranked numeric vector(for GSEA) with names of geneid, specifying the genelist to do enrichment analysis.
`universe`	A character vector, specifying the backgound genelist, default is whole genome.
`method`	One of "ORT"(Over-Representing Test), "GSEA"(Gene Set Enrichment Analysis), "DAVID", "GOstats", and "HGT"(HyperGemetric test), or index from 1 to 5
`type`	Geneset category for testing, KEGG(default).
`organism`	A character, specifying organism, such as "hsa" or "Human"(default), and "mmu" or "Mouse"
`pvalueCutoff`	Pvalue cutoff.
`qvalueCutoff`	Qvalue cutoff.
`pAdjustMethod`	One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
`minGSSize`	Minimal size of each geneSet for testing.
`maxGSSize`	Maximal size of each geneSet for analyzing.
`plotTitle`	Same as 'title' in 'plot'.
`color`	Color of points.

Value

A list, including two items, `gridPlot` and `enrichRes`. `gridPlot` is a ggplot object, and `enrichRes` is a enrichResult instance.

Author(s)

Feizhen Wu

See Also

enrich.GOstats

enrich.DAVID

enrich.GSE

enrich.ORT

enrich.HGT

enrichResult-class

Examples

```
data(geneList, package = "DOSE")
genes <- names(geneList)[1:100]
keggA = enrichment_analysis(genes, method = "HGT", type = "KEGG")
print(keggA$gridPlot)
```

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",
  adjust = "BH", filename = NULL, out.dir = ".", width = 6.5,
  height = 4, ...)
```

Arguments

`beta` Data frame, which contains column of 'group'.

`pvalue` Pvalue cutoff.

`enrich_method` One of "ORT"(Over-Representing Test), "DAVID", "GOstats", and "HGT"(HyperGemetric test).

`organism` A character, specifying organism, such as "hsa" or "Human"(default), and "mmu" or "Mouse"

`adjust` One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

`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

`kegg13` a list record enriched KEGG pathways for Group1&Group3 genes in 9-Square

`kegg14` a list record enriched KEGG pathways for Group1&Group4 genes in 9-Square

`kegg23` a list record enriched KEGG pathways for Group2&Group3 genes in 9-Square

`kegg24` a list record enriched KEGG pathways for Group2&Group4 genes in 9-Square

`bp1` a list record enriched GO BP terms for Group1 genes in 9-Square

`bp2` a list record enriched GO BP terms for Group2 genes in 9-Square

`bp3` a list record enriched GO BP terms for Group3 genes in 9-Square

`bp4` a list record enriched GO BP terms for Group4 genes in 9-Square

`bp13` a list record enriched GO BP terms for Group1&Group3 genes in 9-Square

`bp14` a list record enriched GO BP terms for Group1&Group4 genes in 9-Square

`bp23` a list record enriched GO BP terms for Group2&Group3 genes in 9-Square

`bp24` a list record enriched GO BP terms for Group2&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`

# Read beta score from gene summary table in MAGeCK MLE results

`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 = "Control", treatname = "Treatment",
  organism = "hsa", prefix = "", top = 10, bottom = 10,
  interestGenes = c(), pvalueCutoff = 0.25, adjust = "BH",
  enrich_kegg = "HGT", gsea = FALSE, posControl = NULL,
  scale_cutoff = 1, loess = FALSE, view_allpath = FALSE, outdir = ".")
```

Arguments

`gene_summary` Either a file path or a data frame, which contains columns of 'Gene', `ctrlname`.beta and `treatname`.beta which corresponding to the parameter ctrlname and treatmname.

`ctrlname` A character vector, specifying the name of control samples.

`treatname` A character vector, specifying the name of treatment samples.

`organism` A character, specifying organism, such as "hsa" or "Human"(default), and "mmu" or "Mouse".

`prefix` A character, indicating the prefix of output file name, which can't contain special characters.

`top` An integer, specifying number of top selected genes labeled in rank figure.

`bottom` An integer, specifying number of bottom selected genes labeled in rank figure.

`interestGenes` A character vector, specifying interested genes labeled in rank figure.

`pvalueCutoff` A numeric, specifying pvalue cutoff of enrichment analysis, default 1.

`adjust` One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

`enrich_kegg` One of "HGT"(HyperGemetric test), "ORT"(Over-Representing Test), "DAVID" and "GOstats", specifying enrichment method used for kegg enrichment analysis.

`gsea` Boolean, indicating whether GSEA analysis is needed for positive and negative selection genes.

`posControl` A file path or a character vector, specifying a list of gene entrezid as positive controls used for cell cycle normalization.

`scale_cutoff` Boolean or numeric, whether scale cutoff to whole genome level, or how many standard deviation will be used as cutoff.

`loess` Boolean, whether include loess normalization in the pipeline.

`view_allpath` Boolean, whether output all pathway view figures.

`outdir` Output directory on disk.

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_Data)
## Not run:
  # functional analysis for MAGeCK MLE results
  FluteMLE(MLE_Data, ctrlname=c("D7_R1","D7_R2"), treatname=c("PLX7_R1","PLX7_R2"),
           prefix="BRAF_D7", pvalueCutoff=0.05, 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, prefix = "Test", enrich_kegg = "HGT",
  organism = "hsa", pvalueCutoff = 0.25, adjust = "BH", outdir = ".")
```

Arguments

gene_summary	A file path or a data frame, which has three columns named 'id', 'neg.fdr' and 'pos.fdr'.
prefix	A character, indicating the prefix of output file name.
enrich_kegg	One of "HGT"(HyperGemetric test), "ORT"(Over-Representing Test), "DAVID" and "GOstats", specifying enrichment method used for kegg enrichment analysis.
organism	A character, specifying organism, such as "hsa" or "Human"(default), and "mmu" or "Mouse"
pvalueCutoff	A numeric, specifying pvalue cutoff of enrichment analysis, default 1.
adjust	One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
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_Data)
gene_summary = RRA_Data
## Not run:
    # Run the FluteRRA pipeline
    FluteRRA(gene_summary, prefix="BRAF", 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)
```

HeatmapView *Calculate the similarity between samples and plot heatmap*

Description

Calculate the similarity between samples and plot heatmap

Usage

```
HeatmapView(beta, method = "pearson", breaks = NA, cluster_rows = TRUE,
  cluster_cols = TRUE, legend = TRUE, main = NA, fontsize = 10,
  display_numbers = TRUE, filename = NA, width = NA, height = NA, ...)
```

Arguments

beta	Data frame or matrix, in which each column represents one sample.
method	Character, One of "pearson", "kendall", "spearman", "euclidean", "maximum", "manhattan", "canberra", "binary", or "minkowski".
breaks	The same as that in pheatmap
cluster_rows	The same as that in pheatmap
cluster_cols	The same as that in pheatmap
legend	The same as that in pheatmap
main	The same as that in pheatmap
fontsize	The same as that in pheatmap
display_numbers	The same as that in pheatmap
filename	The same as that in pheatmap
width	The same as that in pheatmap
height	The same as that in pheatmap
...	Other parameters in pheatmap

Value

The same as pheatmap

Author(s)

Wubing Zhang

See Also

pheatmap

Examples

```
data(MLE_Data)
dd = ReadBeta(MLE_Data, organism="hsa")
dd = dd[,3:ncol(dd)]
HeatmapView(dd, method = "pearson")
```

`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

```
gg = data.frame(Label = c("Day0_R1", "Day0_R2", "Day7_R1", "Day7_R2"),
                       Reads = c(62818064, 47289074, 51190401, 58686580))
gg$Reads = gg$Reads / sum(gg$Reads)
IdentBarView(gg, 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 (to check the list, do: data(bods); bods), 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(gse16873.d)
data(demo.paths)
#KEGG view: gene data only
## Not run:
i <- 1
pv.out <- KeggPathwayView(gene.data = gse16873.d[, 1],
       pathway.id = demo.paths$sel.paths[i], 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

```
countSummary = data.frame(Label = c("Day0_R1", "Day0_R2", "Day7_R1", "Day7_R2"),
                           Reads = c(62818064, 47289074, 51190401, 58686580),
                           Mapped = c(39992777, 31709075, 34729858, 37836392))
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_Data)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(MLE_Data, organism="hsa")
MAView(dd, ctrlname = "D7_R1", treatname = "PLX7_R1")
```

`MLE_Data` *Gene summary table in MAGeCK MLE results*

Description

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

Usage

```
data("MLE_Data")
```

Format

A data frame with 17419 observations on the 26 variables.

References

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

Examples

```
data("MLE_Data")
head(MLE_Data)
```

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_Data, organism="hsa")
beta_loess = normalize.loess(beta[,c("D7_R1", "D7_R2", "PLX7_R1", "PLX7_R2")])
```

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, samples = NULL, method = "cell_cycle",
  posControl = NULL, minus = 0.2)
```

Arguments

`beta`	Data frame, in which rows are EntrezID, columns are samples.
`samples`	Character vector, specifying the samples in `beta` to be normalized. If NULL (default), normalize beta score of all samples in `beta`.
`method`	Character, one of 'cell_cycle'(default) and 'loess'.
`posControl`	A file path or a character vector, specifying a list of gene entrezids as positive controls used for cell cycle normalization
`minus`	Numeric, scale for cell cycle normalization. Between 0 and 1.

Details

In CRISPR screens, cells treated with different conditions (e.g., with or without drug) may have different proliferation rates. So we defined a list of core essential genes, which is equally negatively selected between samples with different proliferation rate. Normalization of gene beta scores is performed using these essential genes. `cell_cycle` in MAGeCKFlute normalizes the beta scores of all genes based on the median beta score of essential genes. 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_Data)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(MLE_Data, organism="hsa")
tmp = TransGeneID(rownames(dd), "Symbol", "Entrez")
dd = dd[!(duplicated(tmp)|is.na(tmp)), ]
rownames(dd) = tmp[!(duplicated(tmp)|is.na(tmp))]
samples=c("D7_R1", "D7_R2", "PLX7_R1", "PLX7_R2")
```

```
#Cell Cycle normalization
dd_essential = NormalizeBeta(dd, samples=samples, method=”cell_cycle”)
head(dd_essential)

#Optional loess normalization
dd_loess = NormalizeBeta(dd, samples=samples, 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 = c(), top = 20, bottom = 20,
  cutoff = c(-sd(rankdata), sd(rankdata)), 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` A two-length numeric vector, in which first value is bottom cutoff, and second value is top cutoff.

`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(MLE_Data)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(MLE_Data, organism="hsa")
rankdata = dd$PLX7_R1 - dd$D7_R1
names(rankdata) = rownames(dd)
RankView(rankdata)
```

ReadBeta *Read gene beta scores*

Description

Read gene beta scores from file or data frame

Usage

```
ReadBeta(gene_summary, organism = "hsa")
```

Arguments

gene_summary	A file path or a data frame, data frame, which has columns of 'Gene' and '*|beta'.
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).

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_Data)
dd = ReadBeta(MLE_Data, organism="hsa")
head(dd)
```

ReadRRA *Read MAGeCK-RRA data*

Description

Read pvalue of gene selection from file or data frame

Usage

```
ReadRRA(gene_summary, organism = "hsa")
```

Arguments

gene_summary	A file path or a data frame, which has three columns named 'id', 'neg.fdr' and 'pos.fdr'.
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).

Value

A data frame including four columns, named "Official", "neg.fdr", "pos.fdr" and "ENTREZID".

Author(s)

Wubing Zhang

Examples

```
data(RRA_Data)
dd.rra = ReadRRA(RRA_Data, organism="hsa")
head(dd.rra)
```

RRA_Data *Gene summary data generated by running MAGeCK RRA*

Description

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

Usage

```
data("RRA_Data")
```

Format

A data frame with 17140 observations on 14 variables.

References

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

Examples

```
data("RRA_Data")
head(RRA_Data)
```

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 = 1, 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_Data)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(MLE_Data, organism="hsa")
ScatterView(dd, ctrlname = "D7_R1", treatname = "PLX7_R1")
```

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)
```

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(), scale_cutoff = 1,
  main = NULL, filename = NULL, width = 5, 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.
`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

`ScatterView`

Examples

```
data(MLE_Data)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(MLE_Data, organism="hsa")
SquareView(dd, ctrlname = "D7_R1", treatname = "PLX7_R1")
```

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 = TRUE, 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_Data)
TransGeneID(MLE_Data$Gene[1:10], organism="hsa", useBiomart = FALSE)
TransGeneID(MLE_Data$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_Data)
# Read beta score from gene summary table in MAGeCK MLE results
dd = ReadBeta(MLE_Data, organism="hsa")
ViolinView(dd, samples=c("D7_R1", "D7_R2", "PLX7_R1", "PLX7_R2"))
#or
ViolinView(dd[, c("D7_R1", "D7_R2", "PLX7_R1", "PLX7_R2")])
```

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