

# Package ‘MAGeCKFlute’

September 26, 2024

**Type** Package

**Title** Integrative Analysis Pipeline for Pooled CRISPR Functional Genetic Screens

**Version** 2.8.0

**Date** `` `r Sys.Date() ` ``

**Author** Binbin Wang, Wubing Zhang, Feizhen Wu, Wei Li & X. Shirley Liu

**Maintainer** Wubing Zhang <Watson5bZhang@gmail.com>

**Description** CRISPR (clustered regularly interspaced short palindrome repeats) coupled with nuclease Cas9 (CRISPR/Cas9) screens represent a promising technology to systematically evaluate gene functions. Data analysis for CRISPR/Cas9 screens is a critical process that includes identifying screen hits and exploring biological functions for these hits in downstream analysis. We have previously developed two algorithms, MAGeCK and MAGeCK-VISPR, to analyze CRISPR/Cas9 screen data in various scenarios. These two algorithms allow users to perform quality control, read count generation and normalization, and calculate beta score to evaluate gene selection performance. In downstream analysis, the biological functional analysis is required for understanding biological functions of these identified genes with different screening purposes. Here, We developed MAGeCKFlute for supporting downstream analysis. MAGeCKFlute provides several strategies to remove potential biases within sgRNA-level read counts and gene-level beta scores. The downstream analysis with the package includes identifying essential, non-essential, and target-associated genes, and performing biological functional category analysis, pathway enrichment analysis and protein complex enrichment analysis of these genes. The package also visualizes genes in multiple ways to benefit users exploring screening data. Collectively, MAGeCKFlute enables accurate identification of essential, non-essential, and targeted genes, as well as their related biological functions. This vignette explains the use of the package and demonstrates typical workflows.

**License** GPL (>=3)

**VignetteBuilder** knitr

**Depends** R (>= 4.1)

**Imports** Biobase, gridExtra, ggplot2, ggrepel, grDevices, grid, reshape2, stats, utils, DOSE, clusterProfiler, pathview, enrichplot, msigdb, depmap

**Suggests** biomaRt, BiocStyle, dendextend, graphics, knitr, pheatmap,  
png, scales, sva, BiocManager

**LazyData** TRUE

**NeedsCompilation** no

**biocViews** FunctionalGenomics, CRISPR, PooledScreens, QualityControl,  
Normalization, GeneSetEnrichment, Pathways, Visualization,  
GeneTarget, KEGG

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**git\_url** <https://git.bioconductor.org/packages/MAGeCKFlute>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** cf8b1d4

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-09-25

## Contents

arrangePathview . . . . .	3
BarView . . . . .	5
BatchRemove . . . . .	6
ConsistencyView . . . . .	7
CutoffCalling . . . . .	8
DensityDiffView . . . . .	9
DensityView . . . . .	10
enrich.GSE . . . . .	11
enrich.HGT . . . . .	12
enrich.ORT . . . . .	14
EnrichAB . . . . .	15
EnrichAnalyzer . . . . .	16
EnrichedFilter . . . . .	18
EnrichedGeneView . . . . .	18
EnrichedView . . . . .	20
EnrichSquare . . . . .	21
FluteMLE . . . . .	22
FluteRRA . . . . .	25
getCols . . . . .	27
getGeneAnn . . . . .	28
getOrg . . . . .	28
getOrtAnn . . . . .	29
gsGetter . . . . .	30
hclustView . . . . .	31
HeatmapView . . . . .	32
IdentBarView . . . . .	33

IncorporateDepmap . . . . .	34
loadDepmap . . . . .	35
MapRatesView . . . . .	36
MAView . . . . .	37
noEnrichPlot . . . . .	38
normalize.loess . . . . .	39
NormalizeBeta . . . . .	40
OmitCommonEssential . . . . .	41
RankView . . . . .	42
ReadBeta . . . . .	44
ReadGMT . . . . .	44
ReadRRA . . . . .	45
ReadsgRRA . . . . .	46
reexports . . . . .	46
ResembleDepmap . . . . .	47
retrieve_gs . . . . .	48
ScatterView . . . . .	48
Selector . . . . .	51
sgRankView . . . . .	51
SquareView . . . . .	53
TransGeneID . . . . .	55
ViolinView . . . . .	56
VolcanoView . . . . .	57
writeGMT . . . . .	59

**Index** **60**

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 = NULL,
  sub = NULL,
  organism = "hsa",
  output = ".",
  path.archive = ".",
  kegg.native = TRUE,
  verbose = 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).
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.
verbose	Boolean

**Value**

plot on the current device

**Author(s)**

Wubing Zhang

**Examples**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
colnames(dd)[2:3] = c("Control", "Treatment")
# arrangePathview(dd, c("hsa00534"), title=NULL, sub=NULL, organism="hsa")
```

---

BarView

*Bar plot*

---

## Description

Bar plot

## Usage

```
BarView(  
  df,  
  x = "x",  
  y = "y",  
  fill = "#FC6665",  
  bar.width = 0.8,  
  position = "dodge",  
  dodge.width = 0.8,  
  main = NA,  
  xlab = NULL,  
  ylab = NA,  
  ...  
)
```

## Arguments

df	A data frame.
x	A character, specifying the x-axis.
y	A character, specifying the y-axis.
fill	A character, specifying the fill color.
bar.width	A numeric, specifying the width of bar.
position	"dodge" (default), "stack", "fill".
dodge.width	A numeric, set the width in position_dodge.
main	A character, specifying the figure title.
xlab	A character, specifying the title of x-axis.
ylab	A character, specifying the title of y-axis.
...	Other parameters in geom_bar

## Value

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

## Author(s)

Wubing Zhang

**Examples**

```
mdata = data.frame(group=letters[1:5], count=sample(1:100,5))
BarView(mdata, x = "group", y = "count")
```

---

BatchRemove	<i>Batch effect removal</i>
-------------	-----------------------------

---

**Description**

Batch effect removal

**Usage**

```
BatchRemove(
  mat,
  batchMat,
  log2trans = FALSE,
  pca = TRUE,
  positive = FALSE,
  cluster = FALSE,
  outdir = NULL
)
```

**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.
pca	Boolean, specifying whether return pca plot.
positive	Boolean, specifying whether all values should be positive.
cluster	Boolean, specifying whether perform hierarchical clustering.
outdir	Output directory for hierarchical cluster tree.

**Value**

A list contains 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)
```

---

ConsistencyView

*Visualize the estimate cell cycle compared to control.*

---

## Description

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

## Usage

```
ConsistencyView(
  dat,
  ctrlname,
  treatname,
  main = NULL,
  filename = NULL,
  width = 5,
  height = 4,
  ...
)
```

## Arguments

<code>dat</code>	A data frame.
<code>ctrlname</code>	A character, specifying the names of control samples.
<code>treatname</code>	A character, specifying the names of treatment samples.
<code>main</code>	A character, specifying title.
<code>filename</code>	A character, specifying a file name to create on disk. Set filename to be "NULL", if don't want to save the figure.
<code>width</code>	Numeric, specifying width of figure.
<code>height</code>	Numeric, specifying height of figure.
<code>...</code>	Other available parameters in ggsave.

## Value

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

**Author(s)**

Wubing Zhang

**Examples**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
ConsistencyView(dd, ctrlname = "Pme11_Ctrl", treatname = "Pme11")
```

---

CutoffCalling

*Quantile of normal distribution.*

---

**Description**

Compute cutoff from a normal-distributed vector.

**Usage**

```
CutoffCalling(d, scale = 2)
```

**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	<i>Density plot</i>
-----------------	---------------------

---

**Description**

Plot the distribution of score differences between treatment and control.

**Usage**

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

**Arguments**

<code>dat</code>	A data frame.
<code>ctrlname</code>	A character, specifying the control samples.
<code>treatname</code>	A character, specifying the treatment samples.
<code>main</code>	A character, specifying title.
<code>filename</code>	A character, specifying a file name to create on disk. Set filename to be "NULL", if don't want to save the figure.
<code>width</code>	Numeric, specifying width of figure.
<code>height</code>	Numeric, specifying height of figure.
<code>...</code>	Other parameters in ggsave.

**Value**

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

**Author(s)**

Wubing Zhang

**Examples**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
# Density plot of beta score deviation between control and treatment
DensityDiffView(dd, ctrlname = "Pme11_Ctrl", treatname = "Pme11")
```

---

DensityView

*Density plot*


---

**Description**

Plot the distribution of numeric vectors with the same length.

**Usage**

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

**Arguments**

<code>dat</code>	A data frame.
<code>samples</code>	A character vector, specifying columns in <code>dat</code> for plotting.
<code>main</code>	A character, specifying title.
<code>xlab</code>	A character, specifying title of x-axis.
<code>filename</code>	A character, specifying a file name to create on disk. Set filename to be "NULL", if don't want to save the figure.
<code>width</code>	Numeric, specifying width of figure.
<code>height</code>	Numeric, specifying height of figure.
<code>...</code>	Other available parameters in <code>ggsave</code> .

**Value**

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

**Author(s)**

Wubing Zhang

**See Also**[ViolinView](#)**Examples**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
DensityView(dd, samples=c("Pme11_Ctrl", "Pme11"))
#or
DensityView(dd[, -1])
```

enrich.GSE

*Gene set enrichment analysis***Description**

A universal gene set enrichment analysis tools

**Usage**

```
enrich.GSE(
  geneList,
  keytype = "Symbol",
  type = "GOBP",
  organism = "hsa",
  pvalueCutoff = 1,
  limit = c(2, 100),
  gmtpath = NULL,
  by = "fgsea",
  verbose = TRUE,
  ...
)
```

**Arguments**

geneList	A order ranked numeric vector with geneid as names
keytype	"Entrez", "Ensembl", or "Symbol"
type	Molecular signatures for testing, available datasets include Pathway (KEGG, REACTOME, C2_CP), GO (GOBP, GOCC, GOMF), MSIGDB (C1, C2 (C2_CP (C2_CP_PID, C2_CP_BIOCARTA), C2_CGP), C3 (C3_MIR, C3_TFT), C4, C6, C7, HALLMARK) and Complex (CORUM). Any combination of them are also accessible (e.g. 'GOBP+GOMF+KEGG+REACTOME')

organism	'hsa' or 'mmu'
pvalueCutoff	FDR cutoff
limit	A two-length vector, specifying the minimal and maximal size of gene sets for enrichent analysis
gmtpath	The path to customized gmt file
by	One of 'fgsea' or 'DOSE'
verbose	Boolean
...	Other parameter

**Value**

An enrichResult instance

**Author(s)**

Wubing Zhang

**See Also**

[enrich.HGT](#)

[enrich.ORT](#)

[EnrichAnalyzer](#)

**Examples**

```
data(geneList, package = "DOSE")
## Not run:
  enrichRes = enrich.GSE(geneList, keytype = "entrez")
  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 = "Symbol",
  type = "GOBP",
  organism = "hsa",
  pvalueCutoff = 1,
  limit = c(2, 100),
  universe = NULL,
  gmtpath = NULL,
  verbose = TRUE,
  ...
)
```

**Arguments**

geneList	A numeric vector with gene as names
keytype	"Entrez", "Ensembl", or "Symbol"
type	Molecular signatures for testing, available datasets include Pathway (KEGG, REACTOME, C2_CP), GO (GOBP, GOCC, GOMF), MSIGDB (C1, C2 (C2_CP (C2_CP_PID, C2_CP_BIOCARTA), C2_CGP), C3 (C3_MIR, C3_TFT), C4, C6, C7, HALLMARK) and Complex (CORUM). Any combination of them are also accessible (e.g. 'GOBP+GOMF+KEGG+REACTOME')
organism	'hsa' or 'mmu'
pvalueCutoff	FDR cutoff
limit	A two-length vector, specifying the minimal and maximal size of gene sets for enrichment analysis
universe	A character vector, specifying the background genelist, default is whole genome
gmtpath	The path to customized gmt file
verbose	Boolean
...	Other parameter

**Value**

An 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", keytype = "entrez")
head(slot(enrichRes, "result"))
```

enrich.ORT

*Enrichment analysis using over-representation test***Description**

Enrichment analysis using over-representation test

**Usage**

```
enrich.ORT(
  geneList,
  keytype = "Symbol",
  type = "GOBP",
  organism = "hsa",
  pvalueCutoff = 1,
  limit = c(2, 100),
  universe = NULL,
  gmtpath = NULL,
  verbose = TRUE,
  ...
)
```

**Arguments**

geneList	A numeric vector with gene as names.
keytype	"Entrez" or "Symbol".
type	Molecular signatures for testing, available datasets include Pathway (KEGG, REACTOME, C2_CP), GO (GOBP, GOCC, GOMF), MSIGDB (C1, C2 (C2_CP (C2_CP_PID, C2_CP_BIOCARTA), C2_CGP), C3 (C3_MIR, C3_TFT), C4, C6, C7, HALLMARK) and Complex (CORUM). Any combination of them are also accessible (e.g. 'GOBP+GOMF+KEGG+REACTOME').
organism	'hsa' or 'mmu'.
pvalueCutoff	FDR cutoff.
limit	A two-length vector, specifying the minimal and maximal size of gene sets for enrichment analysis.
universe	A character vector, specifying the background genelist, default is whole genome.
gmtpath	The path to customized gmt file.
verbose	Boolean
...	Other parameter

**Value**

An enrichedResult instance.

**Author(s)**

Wubing Zhang

**See Also**

[enrich.HGT](#)

[enrich.GSE](#)

[EnrichAnalyzer](#)

**Examples**

```
data(geneList, package = "DOSE")
genes <- geneList[1:100]
enrichedRes <- enrich.ORT(genes, keytype = "entrez")
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 Positive and Negative

**Usage**

```
EnrichAB(
  data,
  enrich_method = "HGT",
  top = 10,
  limit = c(2, 100),
  filename = NULL,
  out.dir = ".",
  width = 6.5,
  height = 4,
  verbose = TRUE,
  ...
)
```

**Arguments**

<code>data</code>	A data frame.
<code>enrich_method</code>	One of "ORT" (Over-Representing Test) and "HGT" (HyperGemetric test).
<code>top</code>	An integer, specifying the number of pathways to show.
<code>limit</code>	A two-length vector, specifying the min and max size of pathways for enrichent analysis.
<code>filename</code>	Suffix of output file name.
<code>out.dir</code>	Path to save plot to (combined with filename).
<code>width</code>	As in <code>ggsave</code> .
<code>height</code>	As in <code>ggsave</code> .
<code>verbose</code>	Boolean
<code>...</code>	Other available parameters in <code>ggsave</code> .

**Value**

A list containing enrichment results for each group genes. This list contains eight items, which contain subitems of `gridPlot` and `enrichRes`.

**Author(s)**

Wubing Zhang

---

EnrichAnalyzer

*Enrichment analysis*

---

**Description**

Enrichment analysis

**Usage**

```
EnrichAnalyzer(  
  geneList,  
  keytype = "Symbol",  
  type = "Pathway+GOBP",  
  method = "HGT",  
  organism = "hsa",  
  pvalueCutoff = 1,  
  limit = c(2, 100),  
  universe = NULL,  
  filter = FALSE,  
  gmtpath = NULL,  
  verbose = TRUE  
)
```



**Arguments**

geneList	A numeric vector with gene as names.
keytype	"Entrez" or "Symbol".
type	Molecular signatures for testing, available datasets include Pathway (KEGG, REACTOME, C2_CP), GO (GOBP, GOCC, GOMF), MSIGDB (C1, C2 (C2_CP (C2_CP_PID, C2_CP_BIOCARTA), C2_CGP), C3 (C3_MIR, C3_TFT), C4, C6, C7, HALLMARK) and Complex (CORUM). Any combination of them are also accessible (e.g. 'GOBP+GOMF+KEGG+REACTOME').
method	One of "ORT"(Over-Representing Test), "GSEA"(Gene Set Enrichment Analysis), and "HGT"(HyperGemetric test).
organism	'hsa' or 'mmu'.
pvalueCutoff	FDR cutoff.
limit	A two-length vector (default: c(2, 200)), specifying the minimal and maximal size of gene sets for enrichment analysis.
universe	A character vector, specifying the background genelist, default is whole genome.
filter	Boolean, specifying whether filter out redundancies from the enrichment results.
gmtpath	The path to customized gmt file.
verbose	Boolean

**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")  
## Not run:  
keggA = EnrichAnalyzer(geneList[1:500], keytype = "entrez")  
head(keggA@result)  
  
## End(Not run)
```

---

EnrichedFilter	<i>Simplify the enrichment results based on Jaccard index</i>
----------------	---

---

**Description**

Simplify the enrichment results based on Jaccard index

**Usage**

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

**Arguments**

enrichment	A data frame of enrichment result or an enrichResult object.
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")
## Not run:
  enrichRes <- enrich.HGT(geneList, keytype = "entrez")
  EnrichedFilter(enrichRes)

## End(Not run)
```

---

EnrichedGeneView	<i>Visualize enriched pathways and genes in those pathways</i>
------------------	--

---

**Description**

Visualize enriched pathways and genes in those pathways

**Usage**

```

EnrichedGeneView(
  enrichment,
  geneList,
  rank_by = "p.adjust",
  top = 5,
  bottom = 0,
  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 analysis.
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.
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 label.
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")
## Not run:
  enrichRes <- enrich.GSE(geneList, keytype = "Entrez")
  EnrichedGeneView(enrichment=slot(enrichRes, "result"), geneList, keytype = "Entrez")

## End(Not run)

```

---

 EnrichedView

*View enriched terms*


---

**Description**

Grid plot for enriched terms

**Usage**

```

EnrichedView(
  enrichment,
  rank_by = "pvalue",
  mode = 1,
  subset = NULL,
  top = 0,
  bottom = 0,
  x = "LogFDR",
  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	"pvalue" or "NES", specifying the indices for ranking pathways.
mode	1 or 2.
subset	A vector of pathway ids.
top	An integer, specifying the number of upregulated terms to show.
bottom	An integer, specifying the number of downregulated terms to show.
x	Character, "NES", "LogP", or "LogFDR", 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**

```
data(geneList, package = "DOSE")
## Not run:
  enrichRes = enrich.GSE(geneList, organism="hsa")
  EnrichedView(enrichRes, top = 5, bottom = 5)

## 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,
  id = "GeneID",
  keytype = "Entrez",
  x = "Control",
  y = "Treatment",
  enrich_method = "ORT",
  top = 5,
  limit = c(2, 100),
  filename = NULL,
  out.dir = ".",
  width = 6.5,
  height = 4,
  verbose = TRUE,
  ...
)
```

**Arguments**

beta	Data frame, with columns of "GeneID", "group", and "Diff".
id	A character, indicating the gene column in the data.
keytype	A character, "Symbol" or "Entrez".
x	A character, indicating the x-axis in the 9-square scatter plot.
y	A character, indicating the y-axis in the 9-square scatter plot.
enrich_method	One of "ORT"(Over-Representing Test) and "HGT"(HyperGemetric test).
top	An integer, specifying the number of pathways to show.
limit	A two-length vector, specifying the min and max size of pathways for enrichment 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.
verbose	Boolean.
...	Other available parameters in ggsave.

**Value**

A list containing enrichment results for each group genes. 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

---

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,
  treatname,
  ctrlname = "Depmap",
  keytype = "Symbol",
  organism = "hsa",
  incorporateDepmap = FALSE,
  cell_lines = NA,
  lineages = "All",
  norm_method = "cell_cycle",
  posControl = NULL,
  omitEssential = TRUE,
  top = 10,
  toplabels = NA,
  scale_cutoff = 2,
  limit = c(0, 200),
  enrich_method = "ORT",
  proj = NA,
  width = 10,
  height = 7,
  outdir = ".",
  pathview.top = 4,
  verbose = TRUE
)

```

**Arguments**

<code>gene_summary</code>	A data frame or a file path to gene summary file generated by MAGeCK-MLE.
<code>treatname</code>	A character vector, specifying the names of treatment samples.
<code>ctrlname</code>	A character vector, specifying the names of control samples. If there is no controls in your CRISPR screen, you can specify "Depmap" as ctrlname and set 'incorporateDepmap=TRUE'.
<code>keytype</code>	"Entrez" or "Symbol".
<code>organism</code>	"hsa" or "mmu".
<code>incorporateDepmap</code>	Boolean, indicating whether incorporate Depmap data into analysis.
<code>cell_lines</code>	A character vector, specifying the cell lines in Depmap to be considered.
<code>lineages</code>	A character vector, specifying the lineages in Depmap to be considered.
<code>norm_method</code>	One of "none", "cell_cycle" (default) or "loess".
<code>posControl</code>	A character vector, specifying a list of positive control gene symbols.
<code>omitEssential</code>	Boolean, indicating whether omit common essential genes from the downstream analysis.
<code>top</code>	An integer, specifying the number of top selected genes to be labeled in rank figure and the number of top pathways to be shown.

toplabels	A character vector, specifying interested genes to be labeled in rank figure.
scale_cutoff	Boolean or numeric, specifying how many standard deviation will be used as cutoff.
limit	A two-length vector, specifying the minimal and maximal size of gene sets for enrichment analysis.
enrich_method	One of "ORT"(Over-Representing Test) and "HGT"(HyperGeometric test).
proj	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.
pathview.top	Integer, specifying the number of pathways for pathway visualization.
verbose	Boolean

### 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 output 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/MAGeCKFlute_proj`, which includes a pdf file and many folders. The pdf file 'FluteMLE\_proj\_norm\_method.pdf' is the summary of pipeline results. For each section in this pipeline, figures and useful data are outputted to corresponding subfolders.

- QC: Quality control
- Selection: Positive selection and negative selection.
- Enrichment: Enrichment analysis for positive and negative selection genes.
- PathwayView: Pathway view for top enriched pathways.

### Author(s)

Wubing Zhang

### See Also

[FluteRRA](#)



## Examples

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
## Not run:
# functional analysis for MAGeCK MLE results
FluteMLE(file3, treatname = "Pmel1", ctrlname = "Pmel1_Ctrl", proj = "Pmel1")

## 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 = NULL,
  keytype = "Symbol",
  organism = "hsa",
  incorporateDepmap = FALSE,
  cell_lines = NA,
  lineages = "All",
  omitEssential = TRUE,
  top = 5,
  toplabels = NULL,
  scale_cutoff = 2,
  limit = c(2, 100),
  proj = NA,
  width = 12,
  height = 6,
  outdir = ".",
  verbose = TRUE
)
```

## 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.
keytype	"Entrez" or "Symbol".
organism	"hsa" or "mmu".
incorporateDepmap	Boolean, indicating whether incorporate Depmap data into analysis.

<code>cell_lines</code>	A character vector, specifying the cell lines in Depmap to be considered.
<code>lineages</code>	A character vector, specifying the lineages in Depmap to be considered.
<code>omitEssential</code>	Boolean, indicating whether omit common essential genes from the downstream analysis.
<code>top</code>	An integer, specifying the number of top selected genes to be labeled in rank figure and the number of top pathways to be shown.
<code>toplabels</code>	A character vector, specifying interested genes to be labeled in rank figure.
<code>scale_cutoff</code>	Boolean or numeric, specifying how many standard deviation will be used as cutoff.
<code>limit</code>	A two-length vector, specifying the minimal and maximal size of gene sets for enrichment analysis.
<code>proj</code>	A character, indicating the prefix of output file name.
<code>width</code>	The width of summary pdf in inches.
<code>height</code>	The height of summary pdf in inches.
<code>outdir</code>	Output directory on disk.
<code>verbose</code>	Boolean

### 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/proj_Results`, which includes a pdf file and a folder named 'RRA'.

### Author(s)

Wubing Zhang

### See Also

[FluteMLE](#)

**Examples**

```
file1 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/rra.gene_summary.txt")
file2 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/rra.sgrna_summary.txt")
## Not run:
# Run the FluteRRA pipeline
FluteRRA(file1, file2, proj="Pmel", organism="hsa", incorporateDepmap = FALSE,
scale_cutoff = 1, outdir = "./")

## End(Not run)
```

---

getCols

*Map values to colors*

---

**Description**

Map values to colors

**Usage**

```
getCols(x, palette = 1)
```

**Arguments**

x	A numeric vector.
palette	diverge, rainbow, sequential

**Value**

A vector of colors corresponding to input vector.

**Author(s)**

Wubing Zhang

**Examples**

```
getCols(1:4)
```

getGeneAnn                      *Retrieve gene annotations from the NCBI, HNSC, and Uniprot databases.*

---

**Description**

Retrieve gene annotations from the NCBI, HNSC, and Uniprot databases.

**Usage**

```
getGeneAnn(org = "hsa", update = FALSE)
```

**Arguments**

org                      Character, hsa (default), bta, cfa, mmu, ptr, rno, ssc are optional.  
update                    Boolean, indicating whether download current annotation.

**Value**

A data frame.

**Author(s)**

Wubing Zhang

**Examples**

```
## Not run:  
ann = getGeneAnn("hsa")  
head(ann)  
  
## End(Not run)
```

---

getOrg                      *Get the kegg code of specific mammalia organism.*

---

**Description**

Get the kegg code of specific mammalia organism.

**Usage**

```
getOrg(organism)
```

**Arguments**

organism            Character, KEGG species code, or the common species name. For all potential values check: data(bods); bods. Default org="hsa", and can also be "human" (case insensitive).

**Value**

A list containing three elements:

org                    species  
 pkgannotation package name

**Author(s)**

Wubing Zhang

**Examples**

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

---

getOrtAnn

*Retrieve reference orthologs annotation.*

---

**Description**

Retrieve reference orthologs annotation.

**Usage**

```
getOrtAnn(fromOrg = "mmu", toOrg = "hsa", update = FALSE)
```

**Arguments**

fromOrg            Character, hsa (default), bta, cfa, mmu, ptr, rno, ssc are optional.  
 toOrg              Character, hsa (default), bta, cfa, mmu, ptr, rno, ssc are optional.  
 update            Boolean, indicating whether download recent annotation from NCBI.

**Value**

A data frame.

**Author(s)**

Wubing Zhang

**Examples**

```
## Not run:
ann = getOrtAnn("mmu", "hsa")
head(ann)

## End(Not run)
```

---

gsGetter

---

*Extract pathway annotation from GMT file.*


---

**Description**

Extract pathway annotation from GMT file.

**Usage**

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

**Arguments**

gmtpath	The path to customized gmt file.
type	Molecular signatures for testing, available datasets include Pathway (KEGG, REACTOME, C2_CP:PID, C2_CP:BIOCARTA), GO (GOBP, GOCC, GOMF), MSIGDB (C1, C2 (C2_CP (C2_CP:PID, C2_CP:BIOCARTA), C2_CGP), C3 (C3_MIR, C3_TFT), C4 (C4_CGN, C4_CM), C5 (C5_BP, C5_CC, C5_MF), C6, C7, H) and Complex (CORUM). Any combination of them are also accessible (e.g. 'GOBP+GOMF+KEGG+REACTOME').
limit	A two-length vector, specifying the minimal and maximal size of gene sets to load.
organism	'hsa' or 'mmu'.
update	Boolean, indicating whether update the gene sets from source database.

**Value**

A three-column data frame.

**Author(s)**

Wubing Zhang

**Examples**

```
gene2path = gsGetter(type = "REACTOME+KEGG")
head(gene2path)
```

---

hclustView	<i>Cluster and view cluster tree</i>
------------	--------------------------------------

---

**Description**

Cluster and view cluster tree

**Usage**

```
hclustView(
  d,
  method = "average",
  label_cols = NULL,
  bar_cols = NULL,
  main = NA,
  xlab = NA,
  horiz = TRUE,
  ...
)
```

**Arguments**

<code>d</code>	A dissimilarity structure as produced by <code>dist</code> .
<code>method</code>	The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).
<code>label_cols</code>	A vector to be used as label's colors for the dendrogram.
<code>bar_cols</code>	Either a vector or a matrix, which will be plotted as a colored bar.
<code>main</code>	As in 'plot'.
<code>xlab</code>	As in 'plot'.
<code>horiz</code>	Logical indicating if the dendrogram should be drawn horizontally or not.
<code>...</code>	Arguments to be passed to methods, such as graphical parameters (see <code>par</code> ).

**Value**

Plot figure on open device.

**Author(s)**

Wubing Zhang

**Examples**

```
label_cols = rownames(USArrests)
hclustView(dist(USArrests), label_cols=label_cols, bar_cols=label_cols)
```

HeatmapView

*Draw heatmap***Description**

Draw heatmap

**Usage**

```
HeatmapView(
  mat,
  limit = c(-2, 2),
  na_col = "gray70",
  colPal = rev(colorRampPalette(c("#c12603", "white", "#0073B6"), space = "Lab")(199)),
  filename = NA,
  width = NA,
  height = NA,
  ...
)
```

**Arguments**

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

**Value**

Invisibly a pheatmap object that is a list with components.

**Author(s)**

Wubing Zhang



**Examples**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
gg = cor(dd[,2:ncol(dd)])
HeatmapView(gg, display_numbers = TRUE)
```

---

IdentBarView	<i>Identical bar plot</i>
--------------	---------------------------

---

**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**

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

**Value**

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

**Author(s)**

Wubing Zhang

**Examples**

```
file4 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/countsummary.txt")
countsummary = read.delim(file4, check.names = FALSE)
IdentBarView(countsummary, x="Label", y="Reads")
```

---

IncorporateDepmap

*Incorporate Depmap screen into analysis*

---

**Description**

Incorporate Depmap screen into analysis

**Usage**

```
IncorporateDepmap(
  dd,
  symbol = "id",
  cell_lines = NA,
  lineages = "All",
  na.rm = FALSE
)
```

**Arguments**

dd	A data frame.
symbol	A character, specifying the column name of gene symbols in the data frame.
cell_lines	A character vector, specifying the cell lines for incorporation.
lineages	A character vector, specifying the cancer types for incorporation.
na.rm	Boolean, indicating whether removing NAs from the results.

**Value**

A data frame with Depmap column (average CERES scores across selected cell lines) attached.

**Author(s)**

Wubing Zhang

**Examples**

```
file1 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/rra.gene_summary.txt")
gdata = ReadRRA(file1)
head(gdata)
## Not run:
  gdata = IncorporateDepmap(gdata)
  head(gdata)

## End(Not run)
```

---

loadDepmap

*Load processed Depmap data*

---

**Description**

Load processed Depmap data

**Usage**

LoadDepmap()

**Value**

A list including two elements, one is the Depmap CRISPR data, and the other is the sample annotation data.

**Author(s)**

Wubing Zhang

**Examples**

```
## Not run:
  depmapDat = LoadDepmap()

## End(Not run)
```

---

MapRatesView	<i>View mapping ratio</i>
--------------	---------------------------

---

**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**

```
file4 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/countsummary.txt")
countsummary = read.delim(file4, check.names = FALSE)
MapRatesView(countsummary)
```

**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**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
MAView(dd, ctrlname = "Pmel1_Ctrl", treatname = "Pmel1")
dd2 = NormalizeBeta(dd, method="loess", org = "mmu")
MAView(dd2, ctrlname = "Pmel1_Ctrl", treatname = "Pmel1")
```

---

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 <a href="#">loess</a>
family.loess	Parameter to be passed the function <a href="#">loess</a> . "gaussian" or "symmetric" are acceptable values for this parameter.
...	Any of the options of <code>normalize.loess</code> you would like to modify (described above).

### Value

A matrix similar as mat.

### Author(s)

Wubing Zhang

**See Also**[loess](#)[NormalizeBeta](#)**Examples**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
beta_loess = normalize.loess(dd[,-1])
```

---

NormalizeBeta	<i>Normalize gene beta scores</i>
---------------	-----------------------------------

---

**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,
  org = "hsa"
)
```

**Arguments**

<code>beta</code>	Data frame.
<code>id</code>	An integer specifying the column of gene.
<code>method</code>	Character, one of 'cell_cycle'(default) and 'loess'. or character string giving the name of the table column containing the gene names.
<code>posControl</code>	A character vector, specifying a list of positive control genes.
<code>samples</code>	Character vector, specifying the sample names in <i>beta</i> columns. If NULL (default), take all <i>beta</i> columns as samples.
<code>org</code>	"hsa", "mmu", "bta", "cfa", "ptr", "rno", or "ssc" indicating the organism.



**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**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
## Not run:
#Cell Cycle normalization
dd_essential = NormalizeBeta(dd, method="cell_cycle", org = "mmu")
head(dd_essential)

## End(Not run)
#Optional loess normalization (not recommended)
dd_loess = NormalizeBeta(dd, method="loess")
head(dd_loess)
```

---

OmitCommonEssential    *Omit common essential genes based on depmap data*

---

**Description**

Omit common essential genes based on depmap data

**Usage**

```
OmitCommonEssential(
  dd,
  symbol = "id",
  lineages = "All",
  cell_lines = NULL,
  dependency = -0.5
)
```

**Arguments**

dd	A data frame.
symbol	A character, specifying the column name of gene symbols in the data frame.
lineages	A character vector, specifying the lineages for selecting essential genes.
cell_lines	A character vector, specifying cell lines for selecting essential genes.
dependency	A numeric, specifying the threshold for selecting essential genes.

**Value**

A data frame.

**Author(s)**

Wubing Zhang

**Examples**

```
## Not run:
file1 = file.path(system.file("extdata", package = "MAGeCKFlute"),
                  "testdata/rra.gene_summary.txt")
gdata = ReadRRA(file1)
dim(gdata)
rra.omit = OmitCommonEssential(gdata)
dim(rra.omit)

## End(Not run)
```

---

RankView

*Rank plot*

---

**Description**

Draw the score and rank of genes on a scatter plot.

**Usage**

```
RankView(
  rankdata,
  genelist = NULL,
  decreasing = TRUE,
  top = 5,
  bottom = 5,
  cutoff = 2,
  main = NULL,
  filename = NULL,
  width = 5,
```

```
    height = 4,  
    ...  
  )
```

### Arguments

rankdata	A numeric vector, with gene as names.
genelist	A character vector, specifying genes to be labeled.
decreasing	Boolean, specifying the order of genes to plot.
top	Integer, specifying number of positive genes to be labeled.
bottom	Integer, specifying number of negative genes to be labeled.
cutoff	One numeric value indicating the fold of standard deviation used as cutoff; two number vector, such as c(-1, 1), specifying the exact cutoff for selecting top genes.
main	A character, specifying title.
filename	A character, specifying a file name to create on disk. Set filename to be "NULL", if don't want to save the figure.
width	Numeric, specifying width of figure.
height	Numeric, specifying height of figure.
...	Other available parameters in the function 'geom_text_repel'.

### Value

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

### Author(s)

Wubing Zhang

### Examples

```
file1 = file.path(system.file("extdata", package = "MAGeCKFlute"),  
"testdata/rra.gene_summary.txt")  
gdata = ReadRRA(file1)  
rankdata = gdata$Score  
names(rankdata) = gdata$id  
RankView(rankdata)
```

---

ReadBeta	<i>Read gene beta scores from MAGeCK-MLE results</i>
----------	--

---

**Description**

Read gene beta scores from MAGeCK-MLE results

**Usage**

```
ReadBeta(gene_summary)
```

**Arguments**

gene\_summary    A data frame or a file path to gene summary file generated by MAGeCK-MLE.

**Value**

A data frame, whose first column is Gene and other columns are comparisons.

**Author(s)**

Wubing Zhang

**Examples**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
head(dd)
```

---

ReadGMT	<i>ReadGMT</i>
---------	----------------

---

**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, score = c("lfc", "rra")[1])
```

**Arguments**

gene\_summary    A data frame or a file path to gene summary file generated by MAGeCK-RRA.  
score            "lfc" (default) or "rra", specifying the score type.

**Details**

If the score type is equal to lfc, then LFC will be returned. If the score type is rra, the log10 transformed RRA score will be returned.

**Value**

A data frame including three columns, including "id", "LFC" and "FDR".

**Author(s)**

Wubing Zhang

**Examples**

```
file1 = file.path(system.file("extdata", package = "MAGeCKFlute"),  
"testdata/rra.gene_summary.txt")  
gdata = ReadRRA(file1)  
head(gdata)
```

---

ReadsgRRA	<i>Read sgRNA summary in MAGeCK-RRA results</i>
-----------	---

---

**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**

```
file2 = file.path(system.file("extdata", package = "MAGeCKFlute"),
                  "testdata/rra.sgrna_summary.txt")
sgrra = ReadsgRRA(file2)
head(sgrra)
```

---

reexports	<i>Objects exported from other packages</i>
-----------	---

---

**Description**

These objects are imported from other packages. Follow the links below to see their documentation.

**clusterProfiler** [GSEA](#), [enricher](#)

**enrichplot** [cnetplot](#), [dotplot](#), [emapplot](#), [goplot](#), [gseaplot](#), [gseaplot2](#), [heatplot](#), [ridgeplot](#)

---

ResembleDepmap	<i>Compute the similarity between customized CRISPR screen with Depmap screens</i>
----------------	--

---

**Description**

Compute the similarity between customized CRISPR screen with Depmap screens

**Usage**

```
ResembleDepmap(  
  dd,  
  symbol = "id",  
  score = "Score",  
  lineages = "All",  
  method = c("pearson", "spearman", "kendall")[1]  
)
```

**Arguments**

dd	A data frame.
symbol	A character, specifying the column name of gene symbols in the data frame.
score	A character, specifying the column name of gene essentiality score in the data frame.
lineages	A character vector, specifying the lineages used for common essential gene selection.
method	A character, indicating which correlation coefficient is to be used for the test. One of "pearson", "kendall", or "spearman".

**Value**

A data frame with correlation and test p.value.

**Author(s)**

Wubing Zhang

**Examples**

```
file1 = file.path(system.file("extdata", package = "MAGeCKFlute"),  
"testdata/rra.gene_summary.txt")  
gdata = ReadRRRA(file1)  
## Not run:  
  rra.omit = OmitCommonEssential(gdata)  
  depmap_similarity = ResembleDepmap(rra.omit)  
  head(depmap_similarity)  
  
## End(Not run)
```

retrieve\_gs

*Update genesets from source database*

---

**Description**

Update genesets from source database

**Usage**

```
retrieve_gs(type = c("KEGG", "REACTOME", "CORUM", "GO"), organism = "hsa")
```

**Arguments**

type            A vector of databases, such as KEGG, REACTOME, CORUM, GO.  
organism        'hsa' or 'mmu'.

**Value**

save data to local library.

**Author(s)**

Wubing Zhang

---

ScatterView

*Scatter plot*

---

**Description**

Scatter plot supporting groups.

**Usage**

```
ScatterView(  
  data,  
  x = "x",  
  y = "y",  
  label = 0,  
  model = c("none", "ninesquare", "volcano", "rank")[1],  
  x_cut = NULL,  
  y_cut = NULL,  
  slope = 1,  
  intercept = NULL,  
  auto_cut = FALSE,  
  auto_cut_x = auto_cut,
```



```

    auto_cut_y = auto_cut,
    auto_cut_diag = auto_cut,
    groups = NULL,
    group_col = NULL,
    groupnames = NULL,
    label.top = TRUE,
    top = 0,
    toplabels = NULL,
    display_cut = FALSE,
    color = NULL,
    shape = 16,
    size = 1,
    alpha = 0.6,
    main = NULL,
    xlab = x,
    ylab = y,
    legend.position = "none",
    ...
)

```

### Arguments

<code>data</code>	Data frame.
<code>x</code>	A character, specifying the x-axis.
<code>y</code>	A character, specifying the y-axis.
<code>label</code>	An integer or a character specifying the column used as the label, default value is 0 (row names).
<code>model</code>	One of "none" (default), "ninesquare", "volcano", and "rank".
<code>x_cut</code>	An one or two-length numeric vector, specifying the cutoff used for x-axis.
<code>y_cut</code>	An one or two-length numeric vector, specifying the cutoff used for y-axis.
<code>slope</code>	A numeric value indicating slope of the diagonal cutoff.
<code>intercept</code>	A numeric value indicating intercept of the diagonal cutoff.
<code>auto_cut</code>	Boolean or numeric, specifying how many standard deviation will be used as cutoff.
<code>auto_cut_x</code>	Boolean or numeric, specifying how many standard deviation will be used as cutoff on x-axis.
<code>auto_cut_y</code>	Boolean or numeric, specifying how many standard deviation will be used as cutoff on y-axis
<code>auto_cut_diag</code>	Boolean or numeric, specifying how many standard deviation will be used as cutoff on diagonal.
<code>groups</code>	A character vector specifying groups. Optional groups include "top", "mid", "bottom", "left", "center", "right", "topleft", "topcenter", "topright", "midleft", "midcenter", "midright", "bottomleft", "bottomcenter", "bottomright".
<code>group_col</code>	A vector of colors for specified groups.

groupnames	A vector of group names to show on the legend.
label.top	Boolean, specifying whether label top hits.
top	Integer, specifying the number of top terms in the groups to be labeled.
toplabels	Character vector, specifying terms to be labeled.
display_cut	Boolean, indicating whether display the dashed line of cutoffs.
color	A character, specifying the column name of color in the data frame.
shape	A character, specifying the column name of shape in the data frame.
size	A character, specifying the column name of size in the data frame.
alpha	A numeric, specifying the transparency of the dots.
main	Title of the figure.
xlab	Title of x-axis
ylab	Title of y-axis.
legend.position	Position of legend, "none", "right", "top", "bottom", or a two-length vector indicating the position.
...	Other available parameters in function 'geom_text_repel'.

### Value

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

### Author(s)

Wubing Zhang

### Examples

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
ScatterView(dd, x = "Pm11_Ctrl", y = "Pm11", label = "Gene",
auto_cut = 1, groups = "topright", top = 5, display_cut = TRUE)
ScatterView(dd, x = "Pm11_Ctrl", y = "Pm11", label = "Gene",
auto_cut = 2, model = "ninesquare", top = 5, display_cut = TRUE)
```

---

Selector	<i>Select signatures from candidate list (according to the consistence in most samples).</i>
----------	--

---

**Description**

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

**Usage**

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

**Arguments**

mat	A matrix, each row is candidates (genes), each column is samples.
cutoff	Numeric, specifying the cutoff to define the signatures.
type	Character, ">" or "<".
select	Numeric, specifying the proportion of samples in which signature is selected.

**Value**

An list containing two elements, the first is the selected signature and the 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	<i>View sgRNA rank.</i>
------------	-------------------------

---

**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**

```
file2 = file.path(system.file("extdata", package = "MAGeCKFlute"),
                  "testdata/rra.sgrna_summary.txt")
sgrra = ReadsRRA(file2)
sgRankView(sgrra)
```

---

 SquareView

*Scatter plot showing dots in 9 quadrants*


---

**Description**

Scatter plot showing dots in 9 quadrants

**Usage**

```

SquareView(
  df,
  ctrlname = "Control",
  treatname = "Treatment",
  label = 0,
  label.top = TRUE,
  top = 5,
  genelist = c(),
  x_cut = NULL,
  y_cut = NULL,
  slope = 1,
  intercept = NULL,
  auto_cut = FALSE,
  auto_cut_x = auto_cut,
  auto_cut_y = auto_cut,
  auto_cut_diag = auto_cut,
  groups = c("midleft", "topcenter", "midright", "bottomcenter"),
  groupnames = paste0("Group", 1:length(groups)),
  legend.position = "none",
  main = NULL,
  filename = NULL,
  width = 6,
  height = 4,
  ...
)

```

**Arguments**

<code>df</code>	A data frame.
<code>ctrlname</code>	A character, specifying the names of control samples, of which the average scores will show as the x-axis.
<code>treatname</code>	A character, specifying the name of treatment samples, of which the average scores will show as the y-axis.
<code>label</code>	An integer or a character specifying the column used as the label, default value is 0 (row names).
<code>label.top</code>	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 genes to be labeled.
x_cut	An one or two-length numeric vector, specifying the cutoff used for x-axis.
y_cut	An one or two-length numeric vector, specifying the cutoff used for y-axis.
slope	A numeric value indicating slope of the diagonal cutoff.
intercept	A numeric value indicating intercept of the diagonal cutoff.
auto_cut	Boolean (2-fold SD by default) or numeric, specifying how many standard deviation will be used as cutoff.
auto_cut_x	Boolean (2-fold SD by default) or numeric, specifying how many standard deviation will be used as cutoff on x-axis.
auto_cut_y	Boolean (2-fold SD by default) or numeric, specifying how many standard deviation will be used as cutoff on y-axis
auto_cut_diag	Boolean (2-fold SD by default) or numeric, specifying how many standard deviation will be used as cutoff on 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.
legend.position	Position of the legend.
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**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
SquareView(dd, ctrlname = "Pmel1_Ctrl", treatname = "Pmel1", label = "Gene")
```

---

TransGeneID                      *Gene ID conversion*

---

**Description**

Gene ID conversion

**Usage**

```
TransGeneID(
  genes,
  fromType = "Symbol",
  toType = "Entrez",
  organism = "hsa",
  fromOrg = organism,
  toOrg = organism,
  ensemblHost = "www.ensembl.org",
  unique = TRUE,
  update = FALSE
)
```

**Arguments**

genes	A character vector, input genes to be converted.
fromType	The input ID type, one of "entrez", "symbol"(default), "hgnc", "ensembl", "full-name" and "uniprotswissprot"; you can also input other valid attribute names for biomaRt. Look at the code in examples to check valid attributes.
toType	The output ID type, similar to 'fromType'.
organism	"hsa"(default), "mmu", "bta", "cfa", "ptr", "rno", and "ssc" are optional.
fromOrg	"hsa", "mmu", "bta", "cfa", "ptr", "rno", and "ssc" are optional (Only used when transform gene ids between organisms).
toOrg	"hsa"(default), "mmu", "bta", "cfa", "ptr", "rno", and "ssc" are optional (Only used when transform gene ids between organisms).
ensemblHost	Character, specifying ensembl host, you can use 'listEnsemblArchives()' to show all available Ensembl archives hosts.
unique	Boolean, specifying whether do one-to-one mapping.
update	Boolean, specifying whether update built-in gene annotation (needs network and takes time).

**Value**

A character vector, named by unique input gene ids.

**Author(s)**

Wubing Zhang

## Examples

```
TransGeneID("HLA-A", organism="hsa")
TransGeneID("HLA-A", toType = "uniprot", organism="hsa")
TransGeneID("H2-K1", toType="Symbol", fromOrg = "mmu", toOrg = "hsa")
```

---

ViolinView

*Violin plot*

---

## Description

Violin plot showing the distribution of numeric vectors with the same length.

## Usage

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

## Arguments

<code>dat</code>	A data frame.
<code>samples</code>	A character vector, specifying the columns in the <code>dat</code> for plotting.
<code>main</code>	A character, specifying title.
<code>ylab</code>	A character, specifying title of y-axis.
<code>filename</code>	A character, specifying a file name to create on disk. Set filename to be "NULL", if don't want to save the figure.
<code>width</code>	Numeric, specifying width of figure.
<code>height</code>	Numeric, specifying height of figure.
<code>...</code>	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**

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/mle.gene_summary.txt")
dd = ReadBeta(file3)
ViolinView(dd[, -1])
```

---

VolcanoView

*Volcano View*

---

**Description**

Volcano plot for differential analysis.

**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,
  force = 0.1,
  main = NULL,
  xlab = "log2FC",
  ylab = "-log10(FDR)",
  filename = NULL,
  width = 4,
  height = 2.5,
  ...
)
```

**Arguments**

**df** A data frame.

**x** A character, specifying the x-axis in Volcano figure, 'logFC' (default).

<code>y</code>	A character, specifying the y-axis in Volcano figure, 'adj.P.Val' (default). log10 transformation will be done automatically.
<code>Label</code>	A character, specifying dots to be labeled on the figure.
<code>top</code>	An integer, specifying the number of top significant genes to be labeled.
<code>topnames</code>	A character vector, indicating positive/negative controls to be labeled.
<code>x_cutoff</code>	Numeric, specifying cutoff of the x-axis.
<code>y_cutoff</code>	Numeric, specifying cutoff of the y-axis.
<code>mycolour</code>	A color vector, specifying colors of non-significant, significantly up and down-regulated genes.
<code>alpha</code>	Numeric, parameter in ggplot.
<code>force</code>	Numeric, Parameter for <code>geom_text_repel</code> . Force of repulsion between overlapping text labels.
<code>main</code>	A character, specifying title.
<code>xlab</code>	A character, specifying title of x-axis.
<code>ylab</code>	A character, specifying title of y-axis.
<code>filename</code>	A character, specifying a file name to create on disk. Set filename to be "NULL", if don't want to save the figure.
<code>width</code>	Numeric, specifying width of figure.
<code>height</code>	Numeric, specifying height of figure.
<code>...</code>	Other available parameters in <code>ggsave</code> .

**Value**

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

**Author(s)**

Wubing Zhang

**Examples**

```
file1 = file.path(system.file("extdata", package = "MAGeCKFlute"),
"testdata/rra.gene_summary.txt")
gdata = ReadRRA(file1)
VolcanoView(gdata, x = "Score", y = "FDR", Label = "id")
```

---

writeGMT	<i>Write GMT file</i>
----------	-----------------------

---

**Description**

write data frame to a gmt file

**Usage**

```
writeGMT(gene2path, gmtfile)
```

**Arguments**

gene2path	A data frame. The columns should be Gene, Pathway ID, and Pathway Name.
gmtfile	Path to gmt file.

**Value**

Output gmt file to local folder.

**Author(s)**

Wubing Zhang

**Examples**

```
gene2path = gsGetter(type = "Complex")
# writeGMT(gene2path, "Protein_complex.gmt")
```

# Index

- \* **internal**
  - reexports, [46](#)
- arrangePathview, [3](#)
- BarView, [5](#)
- BatchRemove, [6](#)
- cnetplot, [46](#)
- cnetplot (reexports), [46](#)
- ComBat, [6](#)
- ConsistencyView, [7](#)
- CutoffCalling, [8](#)
- DensityDiffView, [9](#)
- DensityView, [10](#), [57](#)
- dotplot, [46](#)
- dotplot (reexports), [46](#)
- emapplot, [46](#)
- emapplot (reexports), [46](#)
- enrich.GSE, [11](#), [13](#), [15](#), [17](#)
- enrich.HGT, [12](#), [12](#), [15](#), [17](#)
- enrich.ORT, [12](#), [13](#), [14](#), [17](#)
- EnrichAB, [15](#)
- EnrichAnalyzer, [12](#), [13](#), [15](#), [16](#)
- EnrichedFilter, [18](#)
- EnrichedGeneView, [18](#)
- EnrichedView, [20](#), [21](#)
- enricher, [46](#)
- enricher (reexports), [46](#)
- enrichGSE (enrich.GSE), [11](#)
- enrichment (EnrichAnalyzer), [16](#)
- enrichORT (enrich.ORT), [14](#)
- EnrichSquare, [21](#)
- FluteMLE, [22](#), [26](#)
- flutemle (FluteMLE), [22](#)
- FluteRRA, [24](#), [25](#)
- getCols, [27](#)
- getGeneAnn, [28](#)
- getOrg, [28](#)
- getOrtAnn, [29](#)
- goplot, [46](#)
- goplot (reexports), [46](#)
- GSEA, [46](#)
- GSEA (reexports), [46](#)
- gseaplot, [46](#)
- gseaplot (reexports), [46](#)
- gseaplot2, [46](#)
- gseaplot2 (reexports), [46](#)
- gsGetter, [30](#)
- hclustView, [31](#)
- HeatmapView, [32](#)
- heatplot, [46](#)
- heatplot (reexports), [46](#)
- IdentBarView, [33](#)
- IncorporateDepmap, [34](#)
- LoadDepmap (loadDepmap), [35](#)
- loadDepmap, [35](#)
- loess, [39](#), [40](#)
- loess.normalize (normalize.loess), [39](#)
- MapRatesView, [36](#)
- MAView, [37](#)
- noEnrichPlot, [38](#)
- normalize.loess, [39](#)
- NormalizeBeta, [40](#), [40](#)
- normalizebeta (NormalizeBeta), [40](#)
- OmitCommonEssential, [41](#)
- RankView, [42](#)
- rankview (RankView), [42](#)
- ReadBeta, [44](#)
- readbeta (ReadBeta), [44](#)
- ReadGMT, [44](#)

ReadRRA, [45](#)  
readrra (ReadRRA), [45](#)  
ReadsgRRA, [46](#)  
reexports, [46](#)  
ResembleDepmap, [47](#)  
retrieve\_gs, [48](#)  
ridgeplot, [46](#)  
ridgeplot (reexports), [46](#)  
RRApipeline (FluteRRA), [25](#)

ScatterView, [48](#), [54](#)  
Selector, [51](#)  
sgRankView, [51](#)  
SquareView, [53](#)  
squareview (SquareView), [53](#)

TransGeneID, [55](#)

ViolinView, [11](#), [56](#)  
violinview (ViolinView), [56](#)  
VolcanoView, [57](#)

writeGMT, [59](#)