Package 'ClusterGVis'

July 19, 2025

Title One-Step to Cluster and Visualize Gene Expression Data

Version 0.1.4

Description Streamlining the clustering and visualization of time-series gene expression data from RNA-Seq experiments, this tool supports fuzzy c-means and k-means clustering algorithms. It is compatible with outputs from widely-used packages such as 'Seurat', 'Monocle', and 'WGCNA', enabling seamless downstream visualization and analysis. See Lokesh Kumar and Matthias E Futschik (2007) <doi:10.6026/97320630002005> for more details.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.2

Depends R (>= 2.10)

Imports colorRamps,dplyr,e1071,
factoextra,ggplot2,grDevices,grid,magrittr,Matrix,methods,purrr,
reshape2,scales,stats,tibble,SingleCellExperiment

Suggests Biobase, ComplexHeatmap, clusterProfiler,
SummarizedExperiment, TCseq, circlize, igraph, knitr, monocle,
pheatmap, rmarkdown, Seurat, WGCNA, utils, BiocManager

biocViews
clusterProfiler,SummarizedExperiment,ComplexHeatmap,circlize,SingleCellExperiment,TCseq,Biobase

URL https://github.com/junjunlab/ClusterGVis/,
https://junjunlab.github.io/ClusterGvis-manual/

BugReports https://github.com/junjunlab/ClusterGVis/issues

VignetteBuilder knitr

NeedsCompilation no

Author Jun Zhang [aut, cre] (ORCID: <https://orcid.org/0000-0001-7692-9105>)

Maintainer Jun Zhang <3219030654@stu.cpu.edu.cn>

Repository CRAN

Date/Publication 2025-07-19 06:10:02 UTC

Contents

`BEAM_res` *This is a test data for this package test data describtion*

Description

This is a test data for this package test data describtion

Usage

```
BEAM_res
```

Format

An object of class `data.frame` with 47192 rows and 8 columns.

Author(s)

JunZhang

cell_data_set-class *Cell Data Set Class*

Description

Cell Data Set Class

check_dependency *Check and Install Required Packages*

Description

This function checks if the required Bioconductor and CRAN packages are installed. If any of the required packages are not installed, it installs them automatically. This includes Bioconductor packages such as `clusterProfiler`, `SummarizedExperiment`, `ComplexHeatmap`, `SingleCellExperiment`, `TCseq`, `monocle`, and `Biobase`. It also checks for CRAN packages such as `Seurat`, `WGCNA`, `igraph`, `pheatmap`, `circlize`, `e1071`, and `colorRamps`.

Usage

```
check_dependency()
```

Value

NULL This function does not return a value. It installs the packages as needed.

clusterData *Cluster Data Based on Different Methods*

Description

Cluster Data Based on Different Methods

Usage

```
clusterData(
  obj = NULL,
  scaleData = TRUE,
  cluster.method = c("mfuzz", "TCseq", "kmeans", "wgcna"),
  TCseq_params_list = list(),
  object = NULL,
  min.std = 0,
  cluster.num = NULL,
  subcluster = NULL,
  seed = 5201314,
  ...
)
```

Arguments

obj	An input object that can take one of two types: - A **cell_data_set** object for trajectory analysis. - A **matrix** or **data.frame** containing expression data.
scaleData	Logical. Whether to scale the data (e.g., z-score normalization).
cluster.method	Character. Clustering method to use. Options are one of "mfuzz", "TCseq", "kmeans", or "wgcna".
TCseq_params_list	A list of additional parameters passed to the TCseq::timeclust function.
object	A pre-calculated object required when using "wgcna" as the clustering method.
min.std	Numeric. Minimum standard deviation for filtering expression data.
cluster.num	Integer. The number of clusters to identify.
subcluster	A numeric vector of specific cluster IDs to include in the results. If NULL, all clusters are included.
seed	An integer seed for reproducibility in clustering operations.
...	Additional arguments passed to internal functions such as pre_pseudotime_matrix.

Details

Depending on the selected cluster.method, different clustering algorithms are used:

- "mfuzz": Applies Mfuzz soft clustering method, suitable for identifying overlapping clusters.
- "TCseq": Uses TCseq clustering for time-series expression data with support for additional parameters.
- "kmeans": Employs standard k-means clustering via base R's stats::kmeans.
- "wgcna": Leverages pre-calculated WGCNA (Weighted Gene Co-expression Network Analysis) networks.

The function is designed to be flexible, allowing preprocessing (e.g., filtering by min.std), scaling the data (scaleData = TRUE), and generating results compatible with data visualization pipelines.

Value

A list containing the following clustering results:

- **wide.res**: A wide-format data frame with clusters and normalized expression levels.
- **long.res**: A long-format data frame for visualizations, containing cluster information, normalized values, cluster names, and memberships.
- **cluster.list**: A list where each element contains genes belonging to a specific cluster.
- **type**: The clustering method used ("mfuzz", "TCseq", "kmeans", or "wgcna").
- **geneMode**: Currently set to "none" (reserved for future use).
- **geneType**: Currently set to "none" (reserved for future use).

WGCNA Clustering

If the **WGCNA** method is selected, the object parameter must contain a pre-calculated WGCNA network object. This is typically obtained using the WGCNA package functions.

Subsetting Clusters

Use the `subcluster` parameter to focus on specific clusters. Cluster IDs not included in the `subcluster` vector will be excluded from the final results.

Author(s)

JunZhang

This function performs clustering on input data using one of four methods: **mfuzz**, **TCseq**, **kmeans**, or **wgcna**. The clustering results include metadata, normalized data, and cluster memberships.

Examples

```
data("exps")

# kmeans
ck <- clusterData(obj = exps,
                  cluster.method = "kmeans",
                  cluster.num = 8)
```

---

enrichCluster *Perform GO/KEGG Enrichment Analysis for Multiple Clusters*

Description

Perform GO/KEGG Enrichment Analysis for Multiple Clusters

Usage

```
enrichCluster(
  object = NULL,
  type = c("BP", "MF", "CC", "KEGG", "ownSet"),
  TERM2GENE = NULL,
  TERM2NAME = NULL,
  OrgDb = NULL,
  id.trans = TRUE,
  fromType = "SYMBOL",
  toType = c("ENTREZID"),
  readable = TRUE,
  organism = "hsa",
  pvalueCutoff = 0.05,
  topn = 5,
  seed = 5201314,
  add.gene = FALSE,
  use_internal_data = FALSE,
  heatmap.type = c("plot_pseudotime_heatmap2", "plot_genes_branched_heatmap2",
    "plot_multiple_branches_heatmap2"),
  ...
)
```

Arguments

object	An object containing clustering results. This is clusterData object. Alternatively, it can be a CellDataSet object, in which case the function can also visualize pseudotime data.
type	Character. The type of enrichment analysis to perform. Options include: • "BP": Biological Process (GO) • "MF": Molecular Function (GO) • "CC": Cellular Component (GO) • "KEGG": KEGG Pathway analysis • "ownSet": Custom gene set enrichment, requiring TERM2GENE and optionally TERM2NAME.
TERM2GENE	A data frame containing mappings of terms to genes. Required when type = "ownSet". This must be a two-column data frame, where the first column is the term and the second column is the gene.
TERM2NAME	A data frame containing term-to-name mappings. Optional when type = "ownSet". This must also be a two-column data frame, where the first column is the term and the second column is the name.
OrgDb	An organism database object (e.g., org.Hs.eg.db for human or org.Mm.eg.db for mouse), used for GO or KEGG enrichment analysis.
id.trans	Logical. Whether to perform gene ID transformation. Default is TRUE.
fromType	Character. The type of the input gene IDs (e.g., "SYMBOL", "ENSEMBL"). Default is "SYMBOL".
toType	Character. The target ID type for transformation using clusterProfiler::bitr (e.g., "ENTREZID"). Default is "ENTREZID".
readable	Logical. Whether to convert the enrichment result IDs back to a readable format (e.g., SYMBOL). Only applicable for GO and KEGG analysis. Default is TRUE.
organism	Character. The KEGG organism code (e.g., "hsa" for human, "mmu" for mouse). Required when performing KEGG enrichment. Default is "hsa".
pvalueCutoff	Numeric. The p-value cutoff for enriched terms to be included in the results. Default is 0.05.
topn	Integer or vector. The number of top enrichment results to extract. If a single value, it is applied to all clusters. Otherwise, it should match the number of clusters. Default is 5.
seed	Numeric. Seed for random operations to ensure reproducibility. Default is 5201314.
add.gene	Logical. Whether to include the list of genes associated with each enriched term in the results. Default is FALSE.
use_internal_data	Logical, use KEGG.db or latest online KEGG data for enrichKEGG function. Default is FALSE.
heatmap.type	Character. The type of heatmap visualization to use when input data is a CellDataSet object. Options include: • "plot_pseudotime_heatmap2"

- "plot_genes_branched_heatmap2"
- "plot_multiple_branches_heatmap2"

... Additional arguments passed to plot_pseudotime_heatmap2/plot_genes_branched_heatmap2/plot_multip functions.

Value

a data.frame.

Author(s)

JunZhang

This function performs Gene Ontology (GO) or KEGG enrichment analysis, or custom gene set enrichment, on clustered genes. It supports multiple clusters, incorporating cluster-specific results into its analysis.

`exprs` *Generic to access cds count matrix*

Description

Generic to access cds count matrix

Usage

```
exprs(x)
```

Arguments

x	A cell_data_set object.

Value

Count matrix.

Author(s)

https://github.com/cole-trapnell-lab/monocle3

exprs,cell_data_set-method *Method to access cds count matrix*

Description

Method to access cds count matrix

Usage

```
## S4 method for signature 'cell_data_set'
exprs(x)
```

Arguments

x	A cell_data_set object.

Value

Count matrix.

exps *This is a test data for this package test data describtion*

Description

This is a test data for this package test data describtion

Usage

```
exps
```

Format

An object of class `data.frame` with 3767 rows and 6 columns.

Author(s)

Junjun Lao

filter.std modified by Mfuzz filter.std

using filter.std to filter low expression genes

Description

using filter.std to filter low expression genes

Usage

```
filter.std(eset, min.std, visu = TRUE, verbose = TRUE)
```

Arguments

eset	expression matrix, default NULL.
min.std	min stand error, default 0.
visu	whether plot, default FALSE.
verbose	show filter information.

Value

matrix.

getClusters *Determine Optimal Clusters for Gene Expression or Pseudotime Data*

Description

Determine Optimal Clusters for Gene Expression or Pseudotime Data

Usage

```
getClusters(obj = NULL, ...)
```

Arguments

obj	A data object representing the gene expression data or pseudotime data: • If the input is a cell_data_set object (e.g., from Monocle3), the function preprocesses the data using pre_pseudotime_matrix. • If the input is a numeric matrix or a data.frame, it directly uses this data. Default is NULL.
...	Additional arguments passed to the preprocessing function pre_pseudotime_matrix (e.g., assays, normalize, etc.).

Value

A ggplot object visualizing the Elbow plot, where:

- The x-axis represents the number of clusters tested.
- The y-axis represents the WSS for each cluster number.

The optimal cluster number can be visually identified at the "elbow point," where the reduction in WSS diminishes sharply.

a ggplot.

Author(s)

JunZhang

The getClusters function identifies the optimal number of clusters for a given data object. It supports multiple input types, including gene expression matrices and objects such as cell_data_set. The function implements the Elbow method to evaluate within-cluster sum of squares (WSS) across a range of cluster numbers and visualizes the results.

---

net *This is a test data for this package test data describtion*

Description

This is a test data for this package test data describtion

Usage

```
net
```

Format

An object of class list of length 10.

Author(s)

Junjun Lao

`normalized_counts` *Return a size-factor normalized and (optionally) log-transformed expression*

Description

Return a size-factor normalized and (optionally) log-transformed expression

Usage

```
normalized_counts(
  cds,
  norm_method = c("log", "binary", "size_only"),
  pseudocount = 1
)
```

Arguments

`cds`	A CDS object to calculate normalized expression matrix from.
`norm_method`	String indicating the normalization method. Options are "log" (Default), "binary" and "size_only".
`pseudocount`	A pseudocount to add before log transformation. Ignored if norm_method is not "log". Default is 1.

Value

Size-factor normalized, and optionally log-transformed, expression matrix.

Author(s)

https://github.com/cole-trapnell-lab/monocle3

matrix

`plot_genes_branched_heatmap2` *Create a heatmap to demonstrate the bifurcation of gene expression along two branchs which is slightly modified in monocle2*

Description

@description returns a heatmap that shows changes in both lineages at the same time. It also requires that you choose a branch point to inspect. Columns are points in pseudotime, rows are genes, and the beginning of pseudotime is in the middle of the heatmap. As you read from the middle of the heatmap to the right, you are following one lineage through pseudotime. As you read left, the other. The genes are clustered hierarchically, so you can visualize modules of genes that have similar lineage-dependent expression patterns.

Usage

```
plot_genes_branched_heatmap2(
  cds_subset = NULL,
  branch_point = 1,
  branch_states = NULL,
  branch_labels = c("Cell fate 1", "Cell fate 2"),
  cluster_rows = TRUE,
  hclust_method = "ward.D2",
  num_clusters = 6,
  hmcols = NULL,
  branch_colors = c("#979797", "#F05662", "#7990C8"),
  add_annotation_row = NULL,
  add_annotation_col = NULL,
  show_rownames = FALSE,
  use_gene_short_name = TRUE,
  scale_max = 3,
  scale_min = -3,
  norm_method = c("log", "vstExprs"),
  trend_formula = "~sm.ns(Pseudotime, df=3) * Branch",
  return_heatmap = FALSE,
  cores = 1,
  ...
)
```

Arguments

cds_subset	CellDataSet for the experiment (normally only the branching genes detected with branchTest)
branch_point	The ID of the branch point to visualize. Can only be used when reduceDimension is called with method = "DDRTree".
branch_states	The two states to compare in the heatmap. Mutually exclusive with branch_point.
branch_labels	The labels for the branchs.
cluster_rows	Whether to cluster the rows of the heatmap.
hclust_method	The method used by pheatmap to perform hirearchical clustering of the rows.
num_clusters	Number of clusters for the heatmap of branch genes
hmcols	The color scheme for drawing the heatmap.
branch_colors	The colors used in the annotation strip indicating the pre- and post-branch cells.
add_annotation_row	Additional annotations to show for each row in the heatmap. Must be a dataframe with one row for each row in the fData table of cds_subset, with matching IDs.
add_annotation_col	Additional annotations to show for each column in the heatmap. Must be a dataframe with one row for each cell in the pData table of cds_subset, with matching IDs.
show_rownames	Whether to show the names for each row in the table.

use_gene_short_name
: Whether to use the short names for each row. If FALSE, uses row IDs from the fData table.

scale_max
: The maximum value (in standard deviations) to show in the heatmap. Values larger than this are set to the max.

scale_min
: The minimum value (in standard deviations) to show in the heatmap. Values smaller than this are set to the min.

norm_method
: Determines how to transform expression values prior to rendering

trend_formula
: A formula string specifying the model used in fitting the spline curve for each gene/feature.

return_heatmap
: Whether to return the pheatmap object to the user.

cores
: Number of cores to use when smoothing the expression curves shown in the heatmap.

...
: Additional arguments passed to buildBranchCellDataSet

Value

A list of heatmap_matrix (expression matrix for the branch committment), ph (pheatmap heatmap object), annotation_row (annotation data.frame for the row), annotation_col (annotation data.frame for the column).

---

plot_multiple_branches_heatmap2

Create a heatmap to demonstrate the bifurcation of gene expression along multiple branches

---

Description

Create a heatmap to demonstrate the bifurcation of gene expression along multiple branches

Usage

```
plot_multiple_branches_heatmap2(
  cds = NULL,
  branches,
  branches_name = NULL,
  cluster_rows = TRUE,
  hclust_method = "ward.D2",
  num_clusters = 6,
  hmcols = NULL,
  add_annotation_row = NULL,
  add_annotation_col = NULL,
  show_rownames = FALSE,
  use_gene_short_name = TRUE,
  norm_method = c("vstExprs", "log"),
```

```
  scale_max = 3,
  scale_min = -3,
  trend_formula = "~sm.ns(Pseudotime, df=3)",
  return_heatmap = FALSE,
  cores = 1
)
```

Arguments

cds	CellDataSet for the experiment (normally only the branching genes detected with BEAM)
branches	The terminal branches (states) on the developmental tree you want to investigate.
branches_name	Name (for example, cell type) of branches you believe the cells on the branches are associated with.
cluster_rows	Whether to cluster the rows of the heatmap.
hclust_method	The method used by pheatmap to perform hirearchical clustering of the rows.
num_clusters	Number of clusters for the heatmap of branch genes
hmcols	The color scheme for drawing the heatmap.
add_annotation_row	Additional annotations to show for each row in the heatmap. Must be a dataframe with one row for each row in the fData table of cds_subset, with matching IDs.
add_annotation_col	Additional annotations to show for each column in the heatmap. Must be a dataframe with one row for each cell in the pData table of cds_subset, with matching IDs.
show_rownames	Whether to show the names for each row in the table.
use_gene_short_name	Whether to use the short names for each row. If FALSE, uses row IDs from the fData table.
norm_method	Determines how to transform expression values prior to rendering
scale_max	The maximum value (in standard deviations) to show in the heatmap. Values larger than this are set to the max.
scale_min	The minimum value (in standard deviations) to show in the heatmap. Values smaller than this are set to the min.
trend_formula	A formula string specifying the model used in fitting the spline curve for each gene/feature.
return_heatmap	Whether to return the pheatmap object to the user.
cores	Number of cores to use when smoothing the expression curves shown in the heatmap.

Value

A list of heatmap_matrix (expression matrix for the branch committment), ph (pheatmap heatmap object), annotation_row (annotation data.frame for the row), annotation_col (annotation data.frame for the column).

plot_pseudotime_heatmap2 *Plots a pseudotime-ordered, row-centered heatmap which is slightly modified in monocle2*

Description

The function plot_pseudotime_heatmap takes a CellDataSet object (usually containing a only subset of significant genes) and generates smooth expression curves much like plot_genes_in_pseudotime. Then, it clusters these genes and plots them using the pheatmap package. This allows you to visualize modules of genes that co-vary across pseudotime.

Usage

```
plot_pseudotime_heatmap2(
  cds_subset,
  cluster_rows = TRUE,
  hclust_method = "ward.D2",
  num_clusters = 6,
  hmcols = NULL,
  add_annotation_row = NULL,
  add_annotation_col = NULL,
  show_rownames = FALSE,
  use_gene_short_name = TRUE,
  norm_method = c("log", "vstExprs"),
  scale_max = 3,
  scale_min = -3,
  trend_formula = "~sm.ns(Pseudotime, df=3)",
  return_heatmap = FALSE,
  cores = 1
)
```

Arguments

cds_subset	CellDataSet for the experiment (normally only the branching genes detected with branchTest)
cluster_rows	Whether to cluster the rows of the heatmap.
hclust_method	The method used by pheatmap to perform hirearchical clustering of the rows.
num_clusters	Number of clusters for the heatmap of branch genes
hmcols	The color scheme for drawing the heatmap.
add_annotation_row	Additional annotations to show for each row in the heatmap. Must be a dataframe with one row for each row in the fData table of cds_subset, with matching IDs.

add_annotation_col
: Additional annotations to show for each column in the heatmap. Must be a dataframe with one row for each cell in the pData table of cds_subset, with matching IDs.

show_rownames
: Whether to show the names for each row in the table.

use_gene_short_name
: Whether to use the short names for each row. If FALSE, uses row IDs from the fData table.

norm_method
: Determines how to transform expression values prior to rendering

scale_max
: The maximum value (in standard deviations) to show in the heatmap. Values larger than this are set to the max.

scale_min
: The minimum value (in standard deviations) to show in the heatmap. Values smaller than this are set to the min.

trend_formula
: A formula string specifying the model used in fitting the spline curve for each gene/feature.

return_heatmap
: Whether to return the pheatmap object to the user.

cores
: Number of cores to use when smoothing the expression curves shown in the heatmap.

Value

A list of heatmap_matrix (expression matrix for the branch committment), ph (pheatmap heatmap object), annotation_row (annotation data.frame for the row), annotation_col (annotation data.frame for the column).

---

prepareDataFromscRNA *Prepare scRNA Data for clusterGvis Analysis*

Description

This function prepares single-cell RNA sequencing (scRNA-seq) data for differential gene expression analysis. It extracts the expression data for the specified cells and genes, and organizes them into a dataframe format suitable for downstream analysis.

Usage

```
prepareDataFromscRNA(
  object = NULL,
  diffData = NULL,
  showAverage = TRUE,
  cells = NULL,
  group.by = "ident",
  assays = "RNA",
  slot = "data",
  scale.data = TRUE,
```

```
  cluster.order = NULL,
  keep.uniqGene = TRUE,
  sep = "_"
)
```

Arguments

object	an object of class Seurat containing the scRNA-seq data.
diffData	a dataframe containing information about the differential expression analysis which can be output from function FindAllMarkers.
showAverage	a logical indicating whether to show the average gene expression across all cells.
cells	a vector of cell names to extract from the Seurat object. If NULL, all cells will be used.
group.by	a string specifying the grouping variable for differential expression analysis. Default is 'ident', which groups cells by their assigned clusters.
assays	a string or vector of strings specifying the assay(s) to extract from the Seurat object. Default is 'RNA'.
slot	a string specifying the slot name where the assay data is stored in the Seurat object. Default is 'data'.
scale.data	whether do Z-score for expression data, default TRUE.
cluster.order	the celltype orders.
keep.uniqGene	a logical indicating whether to keep only unique gene names. Default is TRUE.
sep	a character string to separate gene and cell names in the output dataframe. Default is "_".

Value

a dataframe containing the expression data for the specified genes and cells, organized in a format suitable for differential gene expression analysis.

---

pre_pseudotime_matrix *Calculate and return a smoothed pseudotime matrix for the given gene list*

Description

This function takes in a monocle3 object and returns a smoothed pseudotime matrix for the given gene list, either in counts or normalized form. The function first matches the gene list with the rownames of the SummarizedExperiment object, and then orders the pseudotime information. The function then uses smooth.spline to apply smoothing to the data. Finally, the function normalizes the data by subtracting the mean and dividing by the standard deviation for each row.

Usage

```
pre_pseudotime_matrix(
  cds_obj = NULL,
  assays = c("counts", "normalized"),
  gene_list = NULL
)
```

Arguments

cds_obj	A monocle3 object
assays	Type of assay to be used for the analysis, either "counts" or "normalized"
gene_list	A vector of gene names

Value

A smoothed pseudotime matrix for the given gene list

---

pseudotime *Generic to extract pseudotime from CDS object*

Description

Generic to extract pseudotime from CDS object

Usage

```
pseudotime(x, reduction_method = "UMAP")
```

Arguments

x	A cell_data_set object.
reduction_method	Reduced dimension to extract pseudotime for.

Value

Pseudotime values.

Author(s)

https://github.com/cole-trapnell-lab/monocle3

`pseudotime,cell_data_set-method` *Method to extract pseudotime from CDS object*

Description

Method to extract pseudotime from CDS object

Usage

```
## S4 method for signature 'cell_data_set'
pseudotime(x, reduction_method = "UMAP")
```

Arguments

`x`	A cell_data_set object.
`reduction_method`	Reduced dimension to extract clusters for.

Value

Pseudotime values.

`sig_gene_names` *This is a test data for this package test data describtion*

Description

This is a test data for this package test data describtion

Usage

```
sig_gene_names
```

Format

An object of class `character` of length 1331.

Author(s)

JunZhang

`size_factors` *Get the size factors from a cds object.*

Description

A wrapper around `colData(cds)$Size_Factor`

Usage

```
size_factors(cds)
```

Arguments

`cds`	A cell_data_set object.

Value

An updated cell_data_set object

`termanno` *This is a test data for this package test data describtion*

Description

This is a test data for this package test data describtion

Usage

```
termanno
```

Format

An object of class `data.frame` with 24 rows and 2 columns.

Author(s)

Junjun Lao

`termanno2` *This is a test data for this package test data describtion*

Description

This is a test data for this package test data describtion

Usage

```
termanno2
```

Format

An object of class `data.frame` with 24 rows and 3 columns.

Author(s)

Junjun Lao

`traverseTree` *traverseTree function*

Description

traverseTree function

Usage

```
traverseTree(g, starting_cell, end_cells)
```

Arguments

`g`	NULL
`starting_cell`	NULL
`end_cells`	NULL

visCluster *using visCluster to visualize cluster results from clusterData and enrichCluster output*

Description

Visualize Clustered Gene Data Using Line Plots and Heatmaps

Usage

```
visCluster(
  object = NULL,
  ht.col.list = list(col_range = c(-2, 0, 2), col_color = c("#08519C", "white",
    "#A50F15")),
  border = TRUE,
  plot.type = c("line", "heatmap", "both"),
  ms.col = c("#0099CC", "grey90", "#CC3333"),
  line.size = 0.1,
  line.col = "grey90",
  add.mline = TRUE,
  mline.size = 2,
  mline.col = "#CC3333",
  ncol = 4,
  ctAnno.col = NULL,
  set.md = "median",
  textbox.pos = c(0.5, 0.8),
  textbox.size = 8,
  panel.arg = c(2, 0.25, 4, "grey90", NA),
  ggplot.panel.arg = c(2, 0.25, 4, "grey90", NA),
  annoTerm.data = NULL,
  annoTerm.mside = "right",
  termAnno.arg = c("grey95", "grey50"),
  add.bar = FALSE,
  bar.width = 8,
  textbar.pos = c(0.8, 0.8),
  go.col = NULL,
  go.size = NULL,
  by.go = "anno_link",
  annoKegg.data = NULL,
  annoKegg.mside = "right",
  keggAnno.arg = c("grey95", "grey50"),
  add.kegg.bar = FALSE,
  kegg.col = NULL,
  kegg.size = NULL,
  by.kegg = "anno_link",
  word_wrap = TRUE,
  add_new_line = TRUE,
```

```
  add.box = FALSE,
  boxcol = NULL,
  box.arg = c(0.1, "grey50"),
  add.point = FALSE,
  point.arg = c(19, "orange", "orange", 1),
  add.line = TRUE,
  line.side = "right",
  markGenes = NULL,
  markGenes.side = "right",
  genes.gp = c("italic", 10, NA),
  term.text.limit = c(10, 18),
  mulGroup = NULL,
  lgd.label = NULL,
  show_row_names = FALSE,
  subgroup.anno = NULL,
  annnoblock.text = TRUE,
  annnoblock.gp = c("white", 8),
  add.sampleanno = TRUE,
  sample.group = NULL,
  sample.col = NULL,
  sample.order = NULL,
  cluster.order = NULL,
  sample.cell.order = NULL,
  HeatmapAnnotation = NULL,
  column.split = NULL,
  cluster_columns = FALSE,
  pseudotime_col = NULL,
  gglist = NULL,
  row_annotation_obj = NULL,
  ...
)
```

Arguments

object	clusterData object, default NULL.
ht.col.list	list of heatmap col_range and col_color, default list(col_range = c(-2, 0, 2),col_color = c("#08519C", "white", "#A50F15")).
border	whether add border for heatmap, default TRUE.
plot.type	the plot type to choose which incuding "line","heatmap" and "both".
ms.col	membership line color form Mfuzz cluster method results, default c('#0099CC','grey90','#CC3333').
line.size	line size for line plot, default 0.1.
line.col	line color for line plot, default "grey90".
add.mline	whether add median line on plot, default TRUE.
mline.size	median line size, default 2.
mline.col	median line color, default "#CC3333".
ncol	the columns for facet plot with line plot, default 4.

ctAnno.col	the heatmap cluster annotation bar colors, default NULL.
set.md	the represent line method on heatmap-line plot(mean/median), default "median".
textbox.pos	the relative position of text in left-line plot, default c(0.5,0.8).
textbox.size	the text size of the text in left-line plot, default 8.
panel.arg	the settings for the left-line panel which are panel size,gap,width,fill and col, default c(2,0.25,4,"grey90",NA).
ggplot.panel.arg	the settings for the ggplot2 object plot panel which are panel size,gap,width,fill and col, default c(2,0.25,4,"grey90",NA).
annoTerm.data	the GO term annotation for the clusters, default NULL.
annoTerm.mside	the wider GO term annotation box side, default "right".
termAnno.arg	the settings for GO term panel annotations which are fill and col, default c("grey95","grey50").
add.bar	whether add bar plot for GO enrichment, default FALSE.
bar.width	the GO enrichment bar width, default 8.
textbar.pos	the barplot text relative position, default c(0.8,0.8).
go.col	the GO term text colors, default NULL.
go.size	the GO term text size(numeric or "pval"), default NULL.
by.go	the GO term text box style("anno_link" or "anno_block"), default "anno_link".
annoKegg.data	the KEGG term annotation for the clusters, default NULL.
annoKegg.mside	the wider KEGG term annotation box side, default "right".
keggAnno.arg	the settings for KEGG term panel annotations which are fill and col, default c("grey95","grey50").
add.kegg.bar	whether add bar plot for KEGG enrichment, default FALSE.
kegg.col	the KEGG term text colors, default NULL.
kegg.size	the KEGG term text size(numeric or "pval"), default NULL.
by.kegg	the KEGG term text box style("anno_link" or "anno_block"), default "anno_link".
word_wrap	whether wrap the text, default TRUE.
add_new_line	whether add new line when text is long, default TRUE.
add.box	whether add boxplot, default FALSE.
boxcol	the box fill colors, default NULL.
box.arg	this is related to boxplot width and border color, default c(0.1,"grey50").
add.point	whether add point, default FALSE.
point.arg	this is related to point shape,fill,color and size, default c(19,"orange","orange",1).
add.line	whether add line, default TRUE.
line.side	the line annotation side, default "right".
markGenes	the gene names to be added on plot, default NULL.
markGenes.side	the gene label side, default "right".
genes.gp	gene labels graphics settings, default c('italic',10,NA).

`term.text.limit`
: the GO term text size limit, default c(10,18).

`mulGroup`
: to draw multiple lines annotation, supply the groups numbers with vector, default NULL.

`lgd.label`
: the lines annotation legend labels, default NULL.

`show_row_names`
: whether to show row names, default FALSE.

`subgroup.anno`
: the sub-cluster for annotation, supply sub-cluster id, default NULL.

`annnoblock.text`
: whether add cluster numbers on right block annotation, default TRUE.

`annnoblock.gp`
: right block annotation text color and size, default c("white",8).

`add.sampleanno`
: whether add column annotation, default TRUE.

`sample.group`
: the column sample groups, default NULL.

`sample.col`
: column annotation colors, default NULL.

`sample.order`
: the orders for column samples, default NULL.

`cluster.order`
: the row cluster orders for user's own defination, default NULL.

`sample.cell.order`
: the celltype order when input is scRNA data and "showAverage = FALSE" for prepareDataFromscRNA.

`HeatmapAnnotation`
: the 'HeatmapAnnotation' object from 'ComplexHeatmap' when you have multiple annotations, default NULL.

`column.split`
: how to split the columns when supply multiple column annotations, default NULL.

`cluster_columns`
: whether cluster the columns, default FALSE.

`pseudotime_col`
: the branch color control for monocle input data.

`gglist`
: a list of ggplot object to annotate each cluster, default NULL.

`row_annotation_obj`
: Row annotation for heatmap, it is a `ComplexHeatmap::rowAnnotation()` object when "markGenes.side" or "line.side" is "right". Otherwise is a list of named vectors.

`...`
: othe aruguments passed by Heatmap fuction.

Details

This function visualizes clustered gene expression data as line plots, heatmaps, or a combination of both, using the `ComplexHeatmap` and `ggplot2` frameworks. Gene annotations, sample annotations, and additional features like custom color schemes and annotations for GO/KEGG terms are supported for visualization.

Value

a ggplot2 or Heatmap object.

Author(s)

JunZhang

Examples

```
data("exps")

# mfuzz
cm <- clusterData(obj = exps,
                  cluster.method = "kmeans",
                  cluster.num = 8)

# plot
visCluster(object = cm,
           plot.type = "line")
```

Index