Package: ClusterGVis (via r-universe)

June 2, 2026

Title One-Step to Cluster and Visualize Gene Expression Data

Version 1.0.0

Description Provides a streamlined workflow for clustering and visualizing gene expression patterns, particularly from time-series RNA-Seq and single-cell experiments. The package is designed to integrate seamlessly within the Bioconductor ecosystem by operating directly on standard data classes such as `SummarizedExperiment` and `SingleCellExperiment`. It implements common clustering algorithms (e.g., k-means, fuzzy c-means) and generates a suite of publication-ready visualizations to explore co-expressed gene modules. Functions are also included to facilitate the visualization of clustering results derived from other popular tools.

License MIT + file LICENSE

Encoding UTF-8

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.5)

Imports colorRamps, dplyr, e1071, factoextra, ggplot2, grDevices, grid, Matrix, methods, purrr, reshape2, scales, stats, tibble, SingleCellExperiment, SummarizedExperiment, igraph, VGAM, scuttle

Suggests Biobase, ComplexHeatmap, clusterProfiler, TCseq, org.Mm.eg.db, circlize, knitr, monocle, pheatmap, rmarkdown, Seurat, WGCNA, utils, BiocManager, S4Vectors, pheatmap, testthat (>= 3.0.0)

biocViews RNASeq,Transcriptomics,Visualization,SingleCell,GeneExpression,Clustering

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

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

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libuv1-dev libxml2-dev zlib1g-dev

Repository https://bioc-release.r-universe.dev

Date/Publication 2026-04-28 13:05:58 UTC

RemoteUrl https://github.com/bioc/ClusterGVis

RemoteRef RELEASE_3_23

RemoteSha cd557acaa4a67fe78e7b9e967093332c29895f66

Contents

`cell_data_set-class` *Cell Data Set Class*

Description

Cell Data Set Class

clusterData *Cluster Data Based on Different Methods*

Description

Cluster Data Based on Different Methods

Usage

```
clusterData(
  obj = NULL,
  scaleData = TRUE,
  clusterMethod = c("mfuzz", "TCseq", "kmeans", "wgcna"),
  TCseqParamsList = list(),
  kmeansParamsList = list(),
  object = NULL,
  minStd = 0,
  clusterNum = NULL,
  subcluster = NULL,
  ...
)
```

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 , or SummarizedExperiment object.
scaleData	Logical. Whether to scale the data (e.g., z-score normalization).
clusterMethod	Character. Clustering method to use. Options are one of "mfuzz", "TCseq", "kmeans", or "wgcna".
TCseqParamsList	A list of additional parameters passed to the TCseq::timeclust function.
kmeansParamsList	A list of additional parameters passed to the stats::kmeans function.
object	A pre-calculated object required when using "wgcna" as the clustering method.
minStd	Numeric. Minimum standard deviation for filtering expression data.
clusterNum	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.
...	Additional arguments passed to internal functions such as pre_pseudotime_matrix.

Details

Depending on the selected `clusterMethod`, 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 `minStd`), 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,
  clusterMethod = "kmeans",
  clusterNum = 8
)
```

DataSet *This is a test data for this package*

Description

This is a test data for this package

This is a test data for this package

This is a test data for this package

This is a test data for this package

This is a test data for this package

This is a test data for this package

This is a test data for this package

This is a test data for this package

Format

A data frame.

A data frame.

A data frame.

A data frame.

A data frame.

A data frame.

A data frame.

A seurat object.

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,
  idTrans = TRUE,
  fromType = "SYMBOL",
  toType = c("ENTREZID"),
  readable = TRUE,
  organism = "hsa",
  pvalueCutoff = 0.05,
  topn = 5,
  addGene = FALSE,
  useInternalData = FALSE,
  heatmapType = 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.
idTrans	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.
addGene	Logical. Whether to include the list of genes associated with each enriched term in the results. Default is FALSE.
useInternalData	Logical, use KEGG.db or latest online KEGG data for enrichKEGG function. Default is FALSE.
heatmapType	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.

Examples

```
library(org.Mm.eg.db)

data("exps")

# kmeans
ck <- clusterData(
  obj = exps,
  clusterMethod = "kmeans",
  clusterNum = 3
)

# enrich for clusters
enrich <- enrichCluster(
  object = ck,
  OrgDb = org.Mm.eg.db,
  type = "BP",
  pvalueCutoff = 0.05,
  topn = 5
)

# check
head(enrich, 3)
```

---

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.

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, minStd, visu = TRUE, verbose = TRUE)
```

Arguments

eset	expression matrix, default NULL.
minStd	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, or SummarizedExperiment object. 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.

Examples

```
data("exps")
getClusters(obj = exps)
```

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).

Examples

```
data("HSMM")
library(pheatmap)
library(VGAM)

# return plot
plot_genes_branched_heatmap2(HSMM,
                             branch_point = 1,
                             num_clusters = 4,
```

```
                                cores = 1,
                                use_gene_short_name = TRUE,
                                show_rownames = TRUE,
                                return_heatmap = TRUE)
```

`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).

pheatmap plot or data frame.

Examples

```
data("HSMM")
library(pheatmap)
library(VGAM)

# return plot
plot_pseudotime_heatmap2(HSMM,
                         num_clusters = 4,
                         cores = 1,
                         show_rownames = TRUE,
                         return_heatmap = TRUE)
```

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,
  groupBy = "ident",
  assays = "RNA",
  slot = "data",
  scaleData = TRUE,
  clusterOrder = NULL,
  keepUniqGene = 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.
groupBy	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'.
scaleData	whether do Z-score for expression data, default TRUE.
clusterOrder	the celltype orders.
keepUniqGene	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.

Examples

```
# Load required libraries
library(Seurat)
library(dplyr)

data("pbmc_subset")

# report only the positive ones
pbmc.markers.all <- Seurat::FindAllMarkers(pbmc_subset,
  only.pos = TRUE,
  min.pct = 0.25,
  logfc.threshold = 0.25
)

# get top 10 genes
pbmc.markers <- pbmc.markers.all |>
  dplyr::group_by(cluster) |>
  dplyr::top_n(n = 20, wt = avg_log2FC)


# prepare data from seurat object
st.data <- prepareDataFromscRNA(
  object = pbmc_subset,
  diffData = pbmc.markers,
  showAverage = TRUE
)
```

---

pseudotime *Generic to extract pseudotime from CDS object*

Description

Generic to extract pseudotime from CDS object

Usage

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

## 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 pseudotime for.

Value

A named numeric vector containing pseudotime values for each cell. The vector has the following properties:

- Names correspond to cell identifiers (column names from the expression matrix)
- Values are numeric pseudotime values representing trajectory progress
- Values typically range from 0 (trajectory start) to maximum trajectory length
- Higher values indicate cells further along the developmental trajectory

Author(s)

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

---

traverseTree *traverseTree function*

Description

traverseTree function

Usage

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

Arguments

g	NULL
starting_cell	NULL
end_cells	NULL

Value

A list

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,
 htColList = list(col_range = c(-2, 0, 2), col_color = c("#08519C", "white", "#A50F15")),
  border = TRUE,
  plotType = c("line", "heatmap", "both"),
  msCol = c("#0099CC", "grey90", "#CC3333"),
  lineSize = 0.1,
  lineCol = "grey90",
  addMline = TRUE,
  mlineSize = 2,
  mlineCol = "#CC3333",
  ncol = 4,
  ctAnnoCol = NULL,
  setMd = "median",
  textboxPos = c(0.5, 0.8),
  textboxSize = 8,
  panelArg = c(2, 0.25, 4, "grey90", NA),
  ggplotPanelArg = c(2, 0.25, 4, "grey90", NA),
  annoTermData = NULL,
  annoTermMside = "right",
  termAnnoArg = c("grey95", "grey50"),
  addBar = FALSE,
  barWidth = 8,
  textbarPos = c(0.8, 0.8),
  goCol = NULL,
  goSize = NULL,
  byGo = "anno_link",
  annoKeggData = NULL,
  annoKeggMside = "right",
  keggAnnoArg = c("grey95", "grey50"),
  addKeggBar = FALSE,
  keggCol = NULL,
  keggSize = NULL,
  byKegg = "anno_link",
  wordWrap = TRUE,
  addNewLine = TRUE,
  addBox = FALSE,
```

```
  boxCol = NULL,
  boxArg = c(0.1, "grey50"),
  addPoint = FALSE,
  pointArg = c(19, "orange", "orange", 1),
  addLine = TRUE,
  lineSide = "right",
  markGenes = NULL,
  markGenesSide = "right",
  genesGp = c("italic", 10, NA),
  termTextLimit = c(10, 18),
  mulGroup = NULL,
  lgdLabel = NULL,
  showRowNames = FALSE,
  subgroupAnno = NULL,
  annnoblockText = TRUE,
  annnoblockGp = c("white", 8),
  addSampleAnno = TRUE,
  sampleGroup = NULL,
  sampleCol = NULL,
  sampleOrder = NULL,
  clusterOrder = NULL,
  sampleCellOrder = NULL,
  heatmapAnnotation = NULL,
  columnSplit = NULL,
  clusterColumns = FALSE,
  pseudotimeCol = NULL,
  gglist = NULL,
  rowAnnotationObj = NULL,
  ...
)
```

Arguments

object	clusterData object, default NULL.
htColList	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.
plotType	the plot type to choose which incuding "line","heatmap" and "both".
msCol	membership line color form Mfuzz cluster method results, default c('#0099CC','grey90','#CC3333').
lineSize	line size for line plot, default 0.1.
lineCol	line color for line plot, default "grey90".
addMline	whether add median line on plot, default TRUE.
mlineSize	median line size, default 2.
mlineCol	median line color, default "#CC3333".
ncol	the columns for facet plot with line plot, default 4.
ctAnnoCol	the heatmap cluster annotation bar colors, default NULL.

setMd	the represent line method on heatmap-line plot(mean/median), default "median".
textboxPos	the relative position of text in left-line plot, default c(0.5,0.8).
textboxSize	the text size of the text in left-line plot, default 8.
panelArg	the settings for the left-line panel which are panel size,gap,width,fill and col, default c(2,0.25,4,"grey90",NA).
ggplotPanelArg	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).
annoTermData	the GO term annotation for the clusters, default NULL.
annoTermMside	the wider GO term annotation box side, default "right".
termAnnoArg	the settings for GO term panel annotations which are fill and col, default c("grey95","grey50").
addBar	whether add bar plot for GO enrichment, default FALSE.
barWidth	the GO enrichment bar width, default 8.
textbarPos	the barplot text relative position, default c(0.8,0.8).
goCol	the GO term text colors, default NULL.
goSize	the GO term text size(numeric or "pval"), default NULL.
byGo	the GO term text box style("anno_link" or "anno_block"), default "anno_link".
annoKeggData	the KEGG term annotation for the clusters, default NULL.
annoKeggMside	the wider KEGG term annotation box side, default "right".
keggAnnoArg	the settings for KEGG term panel annotations which are fill and col, default c("grey95","grey50").
addKeggBar	whether add bar plot for KEGG enrichment, default FALSE.
keggCol	the KEGG term text colors, default NULL.
keggSize	the KEGG term text size(numeric or "pval"), default NULL.
byKegg	the KEGG term text box style("anno_link" or "anno_block"), default "anno_link".
wordWrap	whether wrap the text, default TRUE.
addNewLine	whether add new line when text is long, default TRUE.
addBox	whether add boxplot, default FALSE.
boxCol	the box fill colors, default NULL.
boxArg	this is related to boxplot width and border color, default c(0.1,"grey50").
addPoint	whether add point, default FALSE.
pointArg	this is related to point shape,fill,color and size, default c(19,"orange","orange",1).
addLine	whether add line, default TRUE.
lineSide	the line annotation side, default "right".
markGenes	the gene names to be added on plot, default NULL.
markGenesSide	the gene label side, default "right".
genesGp	gene labels graphics settings, default c('italic',10,NA).
termTextLimit	the GO term text size limit, default c(10,18).

mulGroup	to draw multiple lines annotation, supply the groups numbers with vector, default NULL.
lgdLabel	the lines annotation legend labels, default NULL.
showRowNames	whether to show row names, default FALSE.
subgroupAnno	the sub-cluster for annotation, supply sub-cluster id, default NULL.
annnoblockText	whether add cluster numbers on right block annotation, default TRUE.
annnoblockGp	right block annotation text color and size, default c("white",8).
addSampleAnno	whether add column annotation, default TRUE.
sampleGroup	the column sample groups, default NULL.
sampleCol	column annotation colors, default NULL.
sampleOrder	the orders for column samples, default NULL.
clusterOrder	the row cluster orders for user's own defination, default NULL.
sampleCellOrder	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.
columnSplit	how to split the columns when supply multiple column annotations, default NULL.
clusterColumns	whether cluster the columns, default FALSE.
pseudotimeCol	the branch color control for monocle input data.
gglist	a list of ggplot object to annotate each cluster, default NULL.
rowAnnotationObj	Row annotation for heatmap, it is a `ComplexHeatmap::rowAnnotation()` object when "markGenesSide" or "lineSide" 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,
  clusterMethod = "kmeans",
  clusterNum = 8
)

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

Index