Package ‘CytoSimplex’

April 23, 2025

Type Package

Title Simplex Visualization of Cell Fate Similarity in Single-Cell Data

Version 0.2.0

Date 2025-04-22

Maintainer Yichen Wang <wayichen@umich.edu>

Description Create simplex plots to visualize the similarity between single-cells and selected clusters in a 1-/2-/3-simplex space. Velocity information can be added as an additional layer. See Liu J, Wang Y et al (2023) <doi:10.1093/bioinformatics/btaf119> for more details.

URL https://welch-lab.github.io/CytoSimplex/, https://github.com/welch-lab/CytoSimplex

License GPL-3

Encoding UTF-8

LazyData true

Depends stats, methods, grDevices, R (>= 3.6)

LinkingTo Rcpp, RcppArmadillo

Imports cli, ggplot2, Matrix, plot3D, plotly, RColorBrewer, Rcpp, rlang, viridis

Suggests hdf5r, knitr, magick, patchwork, rmarkdown, Seurat, SeuratObject, SingleCellExperiment, SummarizedExperiment, testthat (>= 3.0.0)

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation yes

Author Yichen Wang [aut, cre] (<https://orcid.org/0000-0003-4347-5199>), Jialin Liu [aut] (<https://orcid.org/0000-0002-9984-7695>), Joshua Welch [cph]

Repository CRAN

Date/Publication 2025-04-23 14:30:02 UTC

Contents

colNormalize 2
plotBinary 4
plotQuaternary 6
plotQuaternary.Seurat 9
plotQuaternary.simMat 12
plotQuaternary.SingleCellExperiment 13
plotTernary 16
plotTernary.Seurat 19
plotTernary.simMat 22
plotTernary.SingleCellExperiment 24
print.quatPlot 27
readH5ADObsNames 28
readH5ADObsVar 29
readH5ADUnsSpMat 30
readVelocytoLoom 30
rnaCluster 32
rnaRaw 32
rnaVelo 33
selectTopFeatures 33
writeQuaternaryGIF 35

Index **37**

`colNormalize` *Normalize each column of the input matrix by the column sum*

Description

Normalize each column of the input matrix by the column sum

Usage

```
colNormalize(x, scaleFactor = NULL, log = FALSE, ...)

## Default S3 method:
colNormalize(x, scaleFactor = NULL, log = FALSE, ...)

## S3 method for class 'dgCMatrix'
colNormalize(x, scaleFactor = NULL, log = FALSE, ...)

## S3 method for class 'Seurat'
colNormalize(
  x,
  scaleFactor = NULL,
  log = FALSE,
  assay = NULL,
  layer = "counts",
  ...
)

## S3 method for class 'SingleCellExperiment'
colNormalize(x, scaleFactor = NULL, log = FALSE, assay.type = "counts", ...)
```

Arguments

x	Feature by observation matrix. Alternatively, `Seurat` object or `SingleCellExperiment` object with raw counts available are also supported.
scaleFactor	Multiplier on normalized data. Default `NULL`.
log	Logical. Whether to take log1p transformation after scaling. Default `FALSE`
...	Additional arguments passed to methods
assay	For "Seurat" method, the specific assay to get data from. Default `NULL` to the default assay.
layer	For "Seurat" method, which layer of the assay to be used. Default `"counts"`.
assay.type	For "SingleCellExperiment" method, the assay type to get data from. Default `"counts"`.

Value

Normalized matrix of the same size

A Seurat object with normalized data in the specified slot of the specified assay.

A SingleCellExperiment object with normalized data in the specified assay. `"normcounts"` if `log = FALSE` and `"logcounts"` if `log = TRUE`.

Examples

```
rnaNorm <- colNormalize(rnaRaw)

# Seurat example
library(Seurat)
srt <- CreateSeuratObject(rnaRaw)
srt <- colNormalize(srt)


# SingleCellExperiment example
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(counts = rnaRaw))
sce <- colNormalize(sce)
```

plotBinary *Create binary plots*

Description

Create binary plots that show similarity between single cells and two selected terminals in a barycentric coordinate. The two vertices are placed at the left and right of a 2D plot where x-axis measures the similarity. Y-axis is jittered for a clear view. A density (histogram) curve is added for indicating the distribution.

See plotTernary manual for more details.

Usage

```
plotBinary(x, ...)

## Default S3 method:
plotBinary(
  x,
  clusterVar,
  vertices,
  features = NULL,
  byCluster = NULL,
  processed = FALSE,
  method = c("euclidean", "cosine", "pearson", "spearman"),
  force = FALSE,
  sigma = 0.08,
  scale = TRUE,
  dotColor = "grey60",
  returnData = FALSE,
  ...
)

## S3 method for class 'Seurat'
plotBinary(
  x,
  layer = "counts",
  assay = NULL,
  clusterVar = NULL,
  processed = FALSE,
  ...
)

## S3 method for class 'SingleCellExperiment'
plotBinary(x, assay.type = "counts", clusterVar = NULL, processed = FALSE, ...)

## S3 method for class 'simMat'
```

```
plotBinary(
  x,
  dotSize = 0.6,
  dotColor = "grey60",
  densLinewidth = 0.8,
  labelColors = c("#3B4992FF", "#EE0000FF"),
  title = NULL,
  ...
)
```

Arguments

`x` Input data. Can be a `matrix` or `dgCMatrix` object with cells as columns, a `Seurat` or `SingleCellExperiment` object. "simMat" method takes intermediate values.

`...` Arguments passed to other methods.

`clusterVar` A vector/factor assigning the cluster variable to each column of the matrix object. For "Seurat" method, `NULL` (default) for `Idents(x)`, or a variable name in `meta.data` slot. For "SingleCellExperiment" method, `NULL` (default) for `colLabels(x)`, or a variable name in `colData` slot.

`vertices` Vector of three unique cluster names that will be used for plotting. Or a named list that groups clusters as three terminal vertices. There must not be any overlap between groups.

`features` Valid matrix row subsetting index to select features for similarity calculation. Default `NULL` uses all available features.

`byCluster` Default `NULL` to generate one plot with all cells. Set `"all"` to split cells in plot by cluster and returns a list of subplots for each cluster as well as the plot including all cells. Otherwise, a vector of cluster names to generate a list of subplots for the specified clusters.

`processed` Logical. Whether the input matrix is already processed. `TRUE` will bypass internal preprocessing and input matrix will be directly used for similarity calculation. Default `FALSE` and raw count input is recommended. If missing in call, using `slot = "counts"` in "Seurat" method or using `assay.type = "counts"` in "SingleCellExperiment" method will force this argument to be `FALSE` and others for `TRUE`.

`method` Similarity calculation method. Default `"euclidean"`. Choose from `"euclidean"`, `"cosine"`, `"pearson"`, `"spearman"`.

`force` Whether to force calculate the similarity when more then 500 features are detected, which is generally not recommended. Default `FALSE`.

`sigma` Gaussian kernel parameter that controls the effect of variance. Only effective when using a distance metric (i.e. `method` is `"euclidian"` or `"cosine"`). Larger value tighten the dot spreading on figure. Default `0.08`.

`scale` Whether to min-max scale the distance matrix by clusters. Default `TRUE`.

`returnData` Logical. Whether to return similarity data instead of generating plot. Default `FALSE`.

`layer` For "Seurat" method, which layer of the assay to be used. Default `"counts"`.

`assay` For "Seurat" method, the specific assay to get data from. Default `NULL` to the default assay.

`assay.type` For "SingleCellExperiment" methods. Which assay to use for calculating the similarity. Default `"counts"`.

`dotSize, dotColor`
Dot aesthetics passed to `geom_point`. Default `0.6` and `"grey60"`.

`densLinewidth` Density plot line aesthetic. Default `0.8`.

`labelColors` Color of the axis lines and vertex labels. Default `c("#3B4992FF", "#EE0000FF")` (blue and red).

`title` Title text of the plot. Default `NULL`.

Value

For 'simMat' method, a ggplot object. For other methods, a ggplot object when `splitCluster = FALSE`, or a list of ggplot objects when `splitCluster = TRUE`.

Examples

```
gene <- selectTopFeatures(rnaRaw, rnaCluster, c("RE", "OS"))
plotBinary(rnaRaw, rnaCluster, c("RE", "OS"), gene)

# Seurat example
library(Seurat)
srt <- CreateSeuratObject(rnaRaw)
Idents(srt) <- rnaCluster
gene <- selectTopFeatures(srt, vertices = c("OS", "RE"))
plotBinary(srt, features = gene, vertices = c("OS", "RE"))


# SingleCellExperiment example
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(counts = rnaRaw))
colLabels(sce) <- rnaCluster
gene <- selectTopFeatures(sce, vertices = c("OS", "RE"))
plotBinary(sce, features = gene, vertices = c("OS", "RE"))
```

`plotQuaternary` *Create quaternary simplex plots*

Description

Create quaternary plots that show similarity between single cells and selected four terminals in a baricentric coordinate.

See `plotTernary` for more details on methodologies.

A dynamic rotating view in a GIF image file can be created with `writeQuaternaryGIF`. Package `magick` must be installed in advance. Linux users may refer to this installation guide.

Usage

```
plotQuaternary(x, ...)

## Default S3 method:
plotQuaternary(
  x,
  clusterVar,
  vertices,
  features = NULL,
  veloGraph = NULL,
  byCluster = NULL,
  processed = FALSE,
  method = c("euclidean", "cosine", "pearson", "spearman"),
  force = FALSE,
  sigma = 0.05,
  scale = TRUE,
  dotColorBy = NULL,
  dotColor = NULL,
  palette = "D",
  direction = 1,
  breaks = NULL,
  legendTitle = NULL,
  returnData = FALSE,
  ...
)
```

Arguments

`x` Input data. Can be a `matrix` or `dgCMatrix` object with cells as columns, a `Seurat` or `SingleCellExperiment` object.

`...` Arguments passed on to `plotQuaternary.simMat`

`nGrid` Number of grids along the x-axis of the tetrahedron. Default `10`.

`radius` Arrow length of unit velocity. Lower this when arrows point outside of the tetrahedron. Default `0.2`.

`dotSize` Size of each dot. Default `0.6` for static figure, and `4` for interactive view.

`labelColors` Colors of the vertex labels. Default `c("#3B4992FF", "#EE0000FF", "#008B45FF", "#631879FF")` (blue, red, green and purple).

`arrowLinewidth` Arrow aesthetics. Default `1.6` for interactive view, `0.6` for static figure.

`arrowAngle,arrowLen` Arrow aesthetics passed to TODOOOO `grid::arrow`. The length of the arrow will be internally converted to unit onject in inches. Default `20` and `0.1`.

`edgeLinewidth` Controls the linewidth of the edges of the tetrahedron. Default `1`.

`vertexLabelSize` Numeric, size of vertex text label relative to default size. Default `1`.

- `title` Title text of the plot. Default `NULL`.
- `titleSize,titleColor` Setting on the main title text. Default `1`, and `"black"`.
- `theta,phi` Numeric scalar. The angles defining the viewing direction. `theta` gives the azimuthal direction and `phi` the colatitude. Default `20` and `0`.
- `interactive` Logical. Whether to display plotly interactive view. Default `TRUE`.

`clusterVar` A vector/factor assigning the cluster variable to each column of the matrix object. For "Seurat" method, `NULL` (default) for `Idents(x)`, or a variable name in `meta.data` slot. For "SingleCellExperiment" method, `NULL` (default) for `colLabels(x)`, or a variable name in `colData` slot.

`vertices` Vector of three unique cluster names that will be used for plotting. Or a named list that groups clusters as three terminal vertices. There must not be any overlap between groups.

`features` Valid matrix row subsetting index to select features for similarity calculation. Default `NULL` uses all available features.

`veloGraph` Cell x cell `dgCMatrix` object containing velocity information. Shows velocity grid-arrow layer when specified. Default `NULL` does not show velocity.

`byCluster` Default `NULL` to generate one plot with all cells. Set `"all"` to split cells in plot by cluster and returns a list of subplots for each cluster as well as the plot including all cells. Otherwise, a vector of cluster names to generate a list of subplots for the specified clusters.

`processed` Logical. Whether the input matrix is already processed. `TRUE` will bypass internal preprocessing and input matrix will be directly used for similarity calculation. Default `FALSE` and raw count input is recommended. If missing in call, using `slot = "counts"` in "Seurat" method or using `assay.type = "counts"` in "SingleCellExperiment" method will force this argument to be `FALSE` and others for `TRUE`.

`method` Similarity calculation method. Default `"euclidean"`. Choose from `"euclidean"`, `"cosine"`, `"pearson"`, `"spearman"`.

`force` Whether to force calculate the similarity when more then 500 features are detected, which is generally not recommended. Default `FALSE`.

`sigma` Gaussian kernel parameter that controls the effect of variance. Only effective when using a distance metric (i.e. `method` is `"euclidian"` or `"cosine"`). Larger values tighten the dot spreading on figure. Default `0.05`.

`scale` Whether to min-max scale the distance matrix by clusters. Default `TRUE`.

`dotColorBy` A vector/factor for coloring dots, can be either categorical (must be character or factor) or continuous. Default `NULL`.

`dotColor` Character vector of color codes. When `dotColorBy` is `NULL`, use one or as many colors as the number of cells. If `dotColorBy` is categorical, specify as many colors as the number of categories in `dotColorBy` or ggplot2 categorical color palette is used by default. If `dotColorBy` is continuous, specify together with `breaks` argument.

`palette` Color palette to use when `dotColorBy` is given. Default `"D"` (viridis) for continuous value and ggplot2 default for categorical value. See detail for alternatives.

`direction` Sets the order of colors in the scale. Default `1` orders as palette default. If `-1`, the order of colors is reversed.

`breaks` Number of breaks for continuous color scale passed to non-interactive "plot3D::scatter3D" call. Default `NULL`.

`legendTitle` Title on the legend/colorbar. Default `NULL` uses `"cluster"` if `dotColorBy` is missing (default); user-end variable expression if `dotColorBy` is directly specified from plotQuaternary.default method; variable name if `dotColorBy` is specified from Seurat or SingleCellExperiment method.

`returnData` Logical. Whether to return similarity and aggregated velocity data if applicable instead of generating plot. Default `FALSE`.

Value

By default, a "plotly" object. When `interactive = FALSE`, a "quatPlot" object when `byCluster` is not specified, or a "list" of "quatPlot" objects when `byCluster` is specified. When `returnData = TRUE`, a list of similarity matrix and aggregated velocity matrix is returned.

See Also

Other plotQuaternary: `plotQuaternary.Seurat()`, `plotQuaternary.SingleCellExperiment()`

Examples

```
gene <- selectTopFeatures(
    x = rnaRaw,
    clusterVar = rnaCluster,
    vertices = c("RE", "OS", "CH", "ORT")
)
plotQuaternary(
    x = rnaRaw,
    clusterVar = rnaCluster,
    vertices = c("RE", "OS", "CH", "ORT"),
    features = gene
)
```

`plotQuaternary.Seurat` *Craete quaternary simplex plot with Seurat object*

Description

Create quaternary plots that show similarity between single cells and selected four terminals in a baricentric coordinate.

See `plotTernary` for more details on methodologies.

A dynamic rotating view in a GIF image file can be created with `writeQuaternaryGIF`. Package `magick` must be installed in advance. Linux users may refer to this installation guide.

Usage

```
## S3 method for class 'Seurat'
plotQuaternary(
  x,
  layer = "counts",
  assay = NULL,
  clusterVar = NULL,
  processed = FALSE,
  dotColorBy = NULL,
  legendTitle = NULL,
  ...
)
```

Arguments

`x` A Seurat object

`layer` Layer in the specified assay to use. Default `"counts"`.

`assay` The assay to get data from. Default `NULL` uses the default assay.

`clusterVar` A variable name in meta.data (`x[[]]`). Default `NULL` uses `Idents(x)`.

`processed` Logical. Whether the input matrix is already processed. `TRUE` will bypass internal preprocessing and input matrix will be directly used for similarity calculation. Default `FALSE` and raw count input is recommended. If missing in call, using `slot = "counts"` in "Seurat" method or using `assay.type = "counts"` in "SingleCellExperiment" method will force this argument to be `FALSE` and others for `TRUE`.

`dotColorBy` A vector/factor for coloring dots, can be either categorical (must be character or factor) or continuous. Default `NULL`.

`legendTitle` Title on the legend/colorbar. Default `NULL` uses `"cluster"` if `dotColorBy` is missing (default); user-end variable expression if `dotColorBy` is directly specified from plotQuaternary.default method; variable name if `dotColorBy` is specified from Seurat or SingleCellExperiment method.

`...` Arguments passed on to `plotQuaternary.default`, `plotQuaternary.simMat`

- `vertices` Vector of three unique cluster names that will be used for plotting. Or a named list that groups clusters as three terminal vertices. There must not be any overlap between groups.
- `features` Valid matrix row subsetting index to select features for similarity calculation. Default `NULL` uses all available features.
- `byCluster` Default `NULL` to generate one plot with all cells. Set `"all"` to split cells in plot by cluster and returns a list of subplots for each cluster as well as the plot including all cells. Otherwise, a vector of cluster names to generate a list of subplots for the specified clusters.
- `veloGraph` Cell x cell `dgCMatrix` object containing velocity information. Shows velocity grid-arrow layer when specified. Default `NULL` does not show velocity.
- `method` Similarity calculation method. Default `"euclidean"`. Choose from `"euclidean"`, `"cosine"`, `"pearson"`, `"spearman"`.

`force` Whether to force calculate the similarity when more then 500 features are detected, which is generally not recommended. Default `FALSE`.

`sigma` Gaussian kernel parameter that controls the effect of variance. Only effective when using a distance metric (i.e. `method` is `"euclidian"` or `"cosine"`). Larger values tighten the dot spreading on figure. Default `0.05`.

`scale` Whether to min-max scale the distance matrix by clusters. Default `TRUE`.

`dotColor` Character vector of color codes. When `dotColorBy` is `NULL`, use one or as many colors as the number of cells. If `dotColorBy` is categorical, specify as many colors as the number of categories in `dotColorBy` or ggplot2 categorical color palette is used by default. If `dotColorBy` is continuous, specify together with `breaks` argument.

`palette` Color palette to use when `dotColorBy` is given. Default `"D"` (viridis) for continuous value and ggplot2 default for categorical value. See detail for alternatives.

`direction` Sets the order of colors in the scale. Default `1` orders as palette default. If `-1`, the order of colors is reversed.

`breaks` Number of breaks for continuous color scale passed to non-interactive "plot3D::scatter3D" call. Default `NULL`.

`returnData` Logical. Whether to return similarity and aggregated velocity data if applicable instead of generating plot. Default `FALSE`.

`nGrid` Number of grids along the x-axis of the tetrahedron. Default `10`.

`radius` Arrow length of unit velocity. Lower this when arrows point outside of the tetrahedron. Default `0.2`.

`dotSize` Size of each dot. Default `0.6` for static figure, and `4` for interactive view.

`labelColors` Colors of the vertex labels. Default `c("#3B4992FF", "#EE0000FF", "#008B45FF", "#631879FF")` (blue, red, green and purple).

`arrowLinewidth` Arrow aesthetics. Default `1.6` for interactive view, `0.6` for static figure.

`arrowAngle,arrowLen` Arrow aesthetics passed to TODOOOO `grid::arrow`. The length of the arrow will be internally converted to unit onject in inches. Default `20` and `0.1`.

`edgeLinewidth` Controls the linewidth of the edges of the tetrahedron. Default `1`.

`vertexLabelSize` Numeric, size of vertex text label relative to default size. Default `1`.

`title` Title text of the plot. Default `NULL`.

`titleSize,titleColor` Setting on the main title text. Default `1`, and `"black"`.

`theta,phi` Numeric scalar. The angles defining the viewing direction. `theta` gives the azimuthal direction and `phi` the colatitude. Default `20` and `0`.

`interactive` Logical. Whether to display plotly interactive view. Default `TRUE`.

Value

By default, a "plotly" object. When `interactive = FALSE`, a "quatPlot" object when `byCluster` is not specified, or a "list" of "quatPlot" objects when `byCluster` is specified. When `returnData =`

TRUE, a list of similarity matrix and aggregated velocity matrix is returned.

See Also

Other plotQuaternary: plotQuaternary(), plotQuaternary.SingleCellExperiment()

Examples

```
# Seurat example
if (requireNamespace("Seurat", quietly = TRUE)) {
  require(Seurat)
  srt <- CreateSeuratObject(rnaRaw)
  Idents(srt) <- rnaCluster
  gene <- selectTopFeatures(srt, vertices = c("OS", "RE", "CH", "ORT"))
  plotQuaternary(srt, features = gene,
                 vertices = c("OS", "RE", "CH", "ORT"))
}
```

---

plotQuaternary.simMat *Create quaternary simplex plot with pre-calculated similarity matrix*

Description

Create quaternary simplex plot with pre-calculated similarity matrix

Usage

```
## S3 method for class 'simMat'
plotQuaternary(
  x,
  veloMat = NULL,
  nGrid = 10,
  radius = 0.2,
  dotSize = NULL,
  colorArg = NULL,
  labelColors = c("#3B4992FF", "#EE0000FF", "#008B45FF", "#631879FF"),
  arrowLinewidth = NULL,
  arrowAngle = 20,
  arrowLen = 0.1,
  vertexLabelSize = NULL,
  edgeLinewidth = 1,
  title = NULL,
  titleSize = 1,
  titleColor = "black",
  theta = 20,
  phi = 0,
  interactive = TRUE,
  ...
)
```

Arguments

`x` simMat object, n cells by 4 vertices, each row summing to 1.

`veloMat` Aggregated velocity matrix. Output of `aggrVeloGraph`.

`nGrid` Number of grids along the x-axis of the tetrahedron. Default `10`.

`radius` Arrow length of unit velocity. Lower this when arrows point outside of the tetrahedron. Default `0.2`.

`dotSize` Size of each dot. Default `0.6` for static figure, and `4` for interactive view.

`colorArg` A "colorArg" object, internally prepared by `plotQuaternary.default`. Default `NULL`.

`labelColors` Colors of the vertex labels. Default `c("#3B4992FF", "#EE0000FF", "#008B45FF", "#631879FF")` (blue, red, green and purple).

`arrowLinewidth` Arrow aesthetics. Default `1.6` for interactive view, `0.6` for static figure.

`arrowAngle`, `arrowLen`
Arrow aesthetics passed to TODOOOO `grid::arrow`. The length of the arrow will be internally converted to unit onject in inches. Default `20` and `0.1`.

`vertexLabelSize`
Numeric, size of vertex text label relative to default size. Default `1`.

`edgeLinewidth` Controls the linewidth of the edges of the tetrahedron. Default `1`.

`title` Title text of the plot. Default `NULL`.

`titleSize`, `titleColor`
Setting on the main title text. Default `1`, and `"black"`.

`theta`, `phi` Numeric scalar. The angles defining the viewing direction. `theta` gives the azimuthal direction and `phi` the colatitude. Default `20` and `0`.

`interactive` Logical. Whether to display plotly interactive view. Default `TRUE`.

`...` Not used

Value

A "quatPlot" object, can be displayed by printing.

`plotQuaternary.SingleCellExperiment`
Create quaternary simplex plot with SingleCellExperiment object

---

Description

Create quaternary plots that show similarity between single cells and selected four terminals in a baricentric coordinate.

See `plotTernary` for more details on methodologies.

A dynamic rotating view in a GIF image file can be created with `writeQuaternaryGIF`. Package `magick` must be installed in advance. Linux users may refer to this installation guide.

Usage

```
## S3 method for class 'SingleCellExperiment'
plotQuaternary(
  x,
  assay.type = "counts",
  clusterVar = NULL,
  processed = FALSE,
  dotColorBy = NULL,
  legendTitle = NULL,
  ...
)
```

Arguments

x A SingleCellExperiment object

assay.type Assay to use for calculating the similarity. Default "counts".

clusterVar A variable name in colData(x). Default NULL uses colLabels(x).

processed Logical. Whether the input matrix is already processed. TRUE will bypass internal preprocessing and input matrix will be directly used for similarity calculation. Default FALSE and raw count input is recommended. If missing in call, using slot = "counts" in "Seurat" method or using assay.type = "counts" in "SingleCellExperiment" method will force this argument to be FALSE and others for TRUE.

dotColorBy A vector/factor for coloring dots, can be either categorical (must be character or factor) or continuous. Default NULL.

legendTitle Title on the legend/colorbar. Default NULL uses "cluster" if dotColorBy is missing (default); user-end variable expression if dotColorBy is directly specified from plotQuaternary.default method; variable name if dotColorBy is specified from Seurat or SingleCellExperiment method.

... Arguments passed on to plotQuaternary.default, plotQuaternary.simMat

- vertices Vector of three unique cluster names that will be used for plotting. Or a named list that groups clusters as three terminal vertices. There must not be any overlap between groups.
- features Valid matrix row subsetting index to select features for similarity calculation. Default NULL uses all available features.
- byCluster Default NULL to generate one plot with all cells. Set "all" to split cells in plot by cluster and returns a list of subplots for each cluster as well as the plot including all cells. Otherwise, a vector of cluster names to generate a list of subplots for the specified clusters.
- veloGraph Cell x cell dgCMatrix object containing velocity information. Shows velocity grid-arrow layer when specified. Default NULL does not show velocity.
- method Similarity calculation method. Default "euclidean". Choose from "euclidean", "cosine", "pearson", "spearman".
- force Whether to force calculate the similarity when more then 500 features are detected, which is generally not recommended. Default FALSE.

`sigma` Gaussian kernel parameter that controls the effect of variance. Only effective when using a distance metric (i.e. `method` is `"euclidian"` or `"cosine"`). Larger values tighten the dot spreading on figure. Default `0.05`.

`scale` Whether to min-max scale the distance matrix by clusters. Default `TRUE`.

`dotColor` Character vector of color codes. When `dotColorBy` is `NULL`, use one or as many colors as the number of cells. If `dotColorBy` is categorical, specify as many colors as the number of categories in `dotColorBy` or ggplot2 categorical color palette is used by default. If `dotColorBy` is continuous, specify together with `breaks` argument.

`palette` Color palette to use when `dotColorBy` is given. Default `"D"` (viridis) for continuous value and ggplot2 default for categorical value. See detail for alternatives.

`direction` Sets the order of colors in the scale. Default `1` orders as palette default. If `-1`, the order of colors is reversed.

`breaks` Number of breaks for continuous color scale passed to non-interactive "plot3D::scatter3D" call. Default `NULL`.

`returnData` Logical. Whether to return similarity and aggregated velocity data if applicable instead of generating plot. Default `FALSE`.

`nGrid` Number of grids along the x-axis of the tetrahedron. Default `10`.

`radius` Arrow length of unit velocity. Lower this when arrows point outside of the tetrahedron. Default `0.2`.

`dotSize` Size of each dot. Default `0.6` for static figure, and `4` for interactive view.

`labelColors` Colors of the vertex labels. Default `c("#3B4992FF", "#EE0000FF", "#008B45FF", "#631879FF")` (blue, red, green and purple).

`arrowLinewidth` Arrow aesthetics. Default `1.6` for interactive view, `0.6` for static figure.

`arrowAngle,arrowLen` Arrow aesthetics passed to TODOOOO `grid::arrow`. The length of the arrow will be internally converted to unit onject in inches. Default `20` and `0.1`.

`edgeLinewidth` Controls the linewidth of the edges of the tetrahedron. Default `1`.

`vertexLabelSize` Numeric, size of vertex text label relative to default size. Default `1`.

`title` Title text of the plot. Default `NULL`.

`titleSize,titleColor` Setting on the main title text. Default `1`, and `"black"`.

`theta,phi` Numeric scalar. The angles defining the viewing direction. `theta` gives the azimuthal direction and `phi` the colatitude. Default `20` and `0`.

`interactive` Logical. Whether to display plotly interactive view. Default `TRUE`.

Value

By default, a "plotly" object. When `interactive = FALSE`, a "quatPlot" object when `byCluster` is not specified, or a "list" of "quatPlot" objects when `byCluster` is specified. When `returnData = TRUE`, a list of similarity matrix and aggregated velocity matrix is returned.

See Also

Other plotQuaternary: plotQuaternary(), plotQuaternary.Seurat()

Examples

```
# SingleCellExperiment example
if (requireNamespace("SingleCellExperiment", quietly = TRUE)) {
  require(SingleCellExperiment)
  sce <- SingleCellExperiment(assays = list(counts = rnaRaw))
  colLabels(sce) <- rnaCluster
  gene <- selectTopFeatures(sce, vertices = c("OS", "RE", "CH", "ORT"))
  plotQuaternary(sce, features = gene,
                 vertices = c("OS", "RE", "CH", "ORT"))
}
```

plotTernary *Create ternary simplex plots*

Description

Create ternary plots that show similarity between single cells and selected three terminals in a ternary baricentric coordinate.

Usage

```
plotTernary(x, ...)

## Default S3 method:
plotTernary(
  x,
  clusterVar,
  vertices,
  features = NULL,
  veloGraph = NULL,
  byCluster = NULL,
  processed = FALSE,
  method = c("euclidean", "cosine", "pearson", "spearman"),
  force = FALSE,
  sigma = 0.08,
  scale = TRUE,
  dotColorBy = NULL,
  dotColor = NULL,
  palette = "D",
  direction = 1,
  breaks = NULL,
  legendTitle = NULL,
  returnData = FALSE,
  ...
)
```

Arguments

x Input data. Can be a `matrix` or `dgCMatrix` object with cells as columns, a `Seurat` or `SingleCellExperiment` object. "simMat" method takes intermediate values.

... Arguments passed on to `plotTernary.simMat`

`title` Title text of the plot. Default `NULL`.

`nGrid` Number of grids along the bottom side of the equilateral triangle. Default `10`.

`radius` Arrow length of unit velocity. Lower this when arrows point outside of the coordinate. Default `0.1`.

`dotSize` Dot aesthetics passed to `geom_point`. Default `0.6` when not interactive, `4` when interactive.

`dotShuffle` Whether to shuffle the order of dots being added to the plot, useful when categorical colors are used and mixing of categories is expected. Default `NULL` does shuffle when `dotColorBy` given is categorical and does not otherwise.

`labelColors` Colors of the axis lines and vertex labels. Default `c("#3B4992FF", "#EE0000FF", "#008B45FF")` (blue, red and green)

`vertexLabelSize` Size of vertex labels. Default `6` when not interactive, `16` when interactive.

`vertexLabelDrift` Position adjustment of the vertex labels, only applied to non-interactive view. Default `0.03`.

`axisBreak` Number of breaks to be labeled along axis. Default `5`.

`axisTextShow` Logical, whether to show axis text. Default `TRUE`.

`axisTextSize` Size of text along each axis break. Default `4` for non-interactive view, `12` for interactive view.

`axisTextDrift` Position adjustment of the axis text, only applied to non-interactive view. Default `0.01`.

`gridLineAlpha` Transparency of background grid lines. Default `0.6`.

`arrowLinewidth` Line width of the velocity arrows. Default `0.25` for non-interactive view, `2` for interactive view.

`arrowAngle` Controls the angle of the arrowhead, only applied to non-interactive view. Default `20`.

`arrowLen` Control length in centimetre from arrow tip to arrow tail, only applied to non-interactive view. Default `0.2`.

`titleSize` Size of title text. Default `14` for non-interactive view, `20` for interactive view.

`equilateral` Logical, whether to always display the triangle as equilateral. Default `TRUE`.

`margin` Margin allowed around of the triangle plotting region when `equilateral = TRUE` `interactive` Logical. Whether to display plotly interactive view. Default `FALSE`.

`clusterVar` A vector/factor assigning the cluster variable to each column of the matrix object. For "Seurat" method, `NULL` (default) for `Idents(x)`, or a variable name in `meta.data` slot. For "SingleCellExperiment" method, `NULL` (default) for `colLabels(x)`, or a variable name in `colData` slot.

`vertices` Vector of three unique cluster names that will be used for plotting. Or a named list that groups clusters as three terminal vertices. There must not be any overlap between groups.

`features` Valid matrix row subsetting index to select features for similarity calculation. Default `NULL` uses all available features.

`veloGraph` Cell x cell `dgCMatrix` object containing velocity information. Shows velocity grid-arrow layer when specified. Default `NULL` does not show velocity.

`byCluster` Default `NULL` to generate one plot with all cells. Set `"all"` to split cells in plot by cluster and returns a list of subplots for each cluster as well as the plot including all cells. Otherwise, a vector of cluster names to generate a list of subplots for the specified clusters.

`processed` Logical. Whether the input matrix is already processed. `TRUE` will bypass internal preprocessing and input matrix will be directly used for similarity calculation. Default `FALSE` and raw count input is recommended. If missing in call, using `slot = "counts"` in "Seurat" method or using `assay.type = "counts"` in "SingleCellExperiment" method will force this argument to be `FALSE` and others for `TRUE`.

`method` Similarity calculation method. Default `"euclidean"`. Choose from `"euclidean"`, `"cosine"`, `"pearson"`, `"spearman"`.

`force` Whether to force calculate the similarity when more then 500 features are detected, which is generally not recommended. Default `FALSE`.

`sigma` Gaussian kernel parameter that controls the effect of variance. Only effective when using a distance metric (i.e. `method` is `"euclidian"` or `"cosine"`). Larger values tighten the dot spreading on figure. Default `0.08`.

`scale` Whether to min-max scale the distance matrix by clusters. Default `TRUE`.

`dotColorBy` A vector/factor for coloring dots, can be either categorical (must be character or factor) or continuous. Default `NULL`.

`dotColor` Character vector of color codes. When `dotColorBy` is `NULL`, use one or as many colors as the number of cells. If `dotColorBy` is categorical, specify as many colors as the number of categories in `dotColorBy` or ggplot2 categorical color palette is used by default. If `dotColorBy` is continuous, specify together with `breaks` argument.

`palette` Color palette to use when `dotColorBy` is given. Default `"D"` (viridis) for continuous value and ggplot2 default for categorical value. See detail for alternatives.

`direction` Sets the order of colors in the scale. Default `1` orders as palette default. If `-1`, the order of colors is reversed.

`breaks` Number of breaks for continuous color scale passed to non-interactive "plot3D::scatter3D" call. Default `NULL`.

`legendTitle` Title on the legend/colorbar. Default `NULL` uses `"cluster"` if `dotColorBy` is missing (default); user-end variable expression if `dotColorBy` is directly specified from plotQuaternary.default method; variable name if `dotColorBy` is specified from Seurat or SingleCellExperiment method.

`returnData` Logical. Whether to return similarity and aggregated velocity data if applicable instead of generating plot. Default `FALSE`.

Details

Argument inheritance - For matrix/dgCMatrix ("default" method), we first calculate the similarity matrix and obtain a "simMat" object. Then the "simMat" method is internally called. For data container objects (e.g. Seurat), we obtain the correct data matrix first and then call the "default" method. The arguments inherits as the flow described above.

The calculation of similarity matrix - The similarity is calculated either by converting a distance metric ("euclidean" or "cosine") with Gaussian kernel, or directly computed with correlation metrics ("pearson" or "spearman"). The centroid of each terminal is obtained first, and the specified metric from each cell to each terminal is calculated. The similarity matrix (n cells by v terminals) is lastly normalized to sum to 1 for each cell, so it becomes a baricentric coordinate.

Value

By default, a "ggplot" object when `byCluster` is not specified, a list of "ggplot" object when `byCluster` is specified. When `interactive = TRUE`, a "plotly" object is returned. When `returnData = TRUE`, a list of similarity matrix and aggregated velocity matrix is returned.

See Also

Other plotTernary: `plotTernary.Seurat()`, `plotTernary.SingleCellExperiment()`

Examples

```
gene <- selectTopFeatures(rnaRaw, rnaCluster, c("OS", "RE", "CH"))
plotTernary(rnaRaw, rnaCluster, c("OS", "RE", "CH"), gene)
```

`plotTernary.Seurat` *Create ternary simplex plot with Seurat objects*

Description

Create ternary plots that show similarity between single cells and selected three terminals in a ternary baricentric coordinate.

Usage

```
## S3 method for class 'Seurat'
plotTernary(
  x,
  layer = "counts",
  assay = NULL,
  clusterVar = NULL,
  processed = FALSE,
  dotColorBy = NULL,
  legendTitle = NULL,
  ...
)
```

Arguments

`x` A Seurat object

`layer` Layer in the specified assay to use. Default `"counts"`.

`assay` The assay to get data from. Default `NULL` uses the default assay.

`clusterVar` A variable name in meta.data (`x[[]]`). Default `NULL` uses `Idents(x)`.

`processed` Logical. Whether the input matrix is already processed. `TRUE` will bypass internal preprocessing and input matrix will be directly used for similarity calculation. Default `FALSE` and raw count input is recommended. If missing in call, using `slot = "counts"` in "Seurat" method or using `assay.type = "counts"` in "SingleCellExperiment" method will force this argument to be `FALSE` and others for `TRUE`.

`dotColorBy` A vector/factor for coloring dots, can be either categorical (must be character or factor) or continuous. Default `NULL`.

`legendTitle` Title on the legend/colorbar. Default `NULL` uses `"cluster"` if `dotColorBy` is missing (default); user-end variable expression if `dotColorBy` is directly specified from plotQuaternary.default method; variable name if `dotColorBy` is specified from Seurat or SingleCellExperiment method.

`...` Arguments passed on to `plotTernary.default`, `plotTernary.simMat`

- `vertices` Vector of three unique cluster names that will be used for plotting. Or a named list that groups clusters as three terminal vertices. There must not be any overlap between groups.
- `features` Valid matrix row subsetting index to select features for similarity calculation. Default `NULL` uses all available features.
- `byCluster` Default `NULL` to generate one plot with all cells. Set `"all"` to split cells in plot by cluster and returns a list of subplots for each cluster as well as the plot including all cells. Otherwise, a vector of cluster names to generate a list of subplots for the specified clusters.
- `veloGraph` Cell x cell `dgCMatrix` object containing velocity information. Shows velocity grid-arrow layer when specified. Default `NULL` does not show velocity.
- `method` Similarity calculation method. Default `"euclidean"`. Choose from `"euclidean"`, `"cosine"`, `"pearson"`, `"spearman"`.

`force` Whether to force calculate the similarity when more then 500 features are detected, which is generally not recommended. Default `FALSE`.

`sigma` Gaussian kernel parameter that controls the effect of variance. Only effective when using a distance metric (i.e. `method` is `"euclidian"` or `"cosine"`). Larger values tighten the dot spreading on figure. Default `0.08`.

`scale` Whether to min-max scale the distance matrix by clusters. Default `TRUE`.

`dotColor` Character vector of color codes. When `dotColorBy` is `NULL`, use one or as many colors as the number of cells. If `dotColorBy` is categorical, specify as many colors as the number of categories in `dotColorBy` or ggplot2 categorical color palette is used by default. If `dotColorBy` is continuous, specify together with `breaks` argument.

`palette` Color palette to use when `dotColorBy` is given. Default `"D"` (viridis) for continuous value and ggplot2 default for categorical value. See detail for alternatives.

`direction` Sets the order of colors in the scale. Default `1` orders as palette default. If `-1`, the order of colors is reversed.

`breaks` Number of breaks for continuous color scale passed to non-interactive "plot3D::scatter3D" call. Default `NULL`.

`returnData` Logical. Whether to return similarity and aggregated velocity data if applicable instead of generating plot. Default `FALSE`.

`title` Title text of the plot. Default `NULL`.

`nGrid` Number of grids along the bottom side of the equilateral triangle. Default `10`.

`radius` Arrow length of unit velocity. Lower this when arrows point outside of the coordinate. Default `0.1`.

`dotSize` Dot aesthetics passed to `geom_point`. Default `0.6` when not interactive, `4` when interactive.

`dotShuffle` Whether to shuffle the order of dots being added to the plot, useful when categorical colors are used and mixing of categories is expected. Default `NULL` does shuffle when `dotColorBy` given is categorical and does not otherwise.

`labelColors` Colors of the axis lines and vertex labels. Default `c("#3B4992FF", "#EE0000FF", "#008B45FF")` (blue, red and green)

`vertexLabelSize` Size of vertex labels. Default `6` when not interactive, `16` when interactive.

`vertexLabelDrift` Position adjustment of the vertex labels, only applied to non-interactive view. Default `0.03`.

`axisBreak` Number of breaks to be labeled along axis. Default `5`.

`axisTextShow` Logical, whether to show axis text. Default `TRUE`.

`axisTextSize` Size of text along each axis break. Default `4` for non-interactive view, `12` for interactive view.

`axisTextDrift` Position adjustment of the axis text, only applied to non-interactive view. Default `0.01`.

`gridLineAlpha` Transparency of background grid lines. Default `0.6`.

`arrowLinewidth` Line width of the velocity arrows. Default `0.25` for non-interactive view, `2` for interactive view.

- `arrowAngle` Controls the angle of the arrowhead, only applied to non-interactive view. Default `20`.
- `arrowLen` Control length in centimetre from arrow tip to arrow tail, only applied to non-interactive view. Default `0.2`.
- `titleSize` Size of title text. Default `14` for non-interactive view, `20` for interactive view.
- `equilateral` Logical, whether to always display the triangle as equilateral. Default `TRUE`.
- `margin` Margin allowed around of the triangle plotting region when `equilateral = TRUE`
- `interactive` Logical. Whether to display plotly interactive view. Default `FALSE`.

Value

By default, a "ggplot" object when `byCluster` is not specified, a list of "ggplot" object when `byCluster` is specified. When `interactive = TRUE`, a "plotly" object is returned. When `returnData = TRUE`, a list of similarity matrix and aggregated velocity matrix is returned.

See Also

Other plotTernary: `plotTernary()`, `plotTernary.SingleCellExperiment()`

Examples

```
# Seurat example
library(Seurat)
srt <- CreateSeuratObject(rnaRaw)
Idents(srt) <- rnaCluster
gene <- selectTopFeatures(srt, vertices = c("OS", "RE", "CH"))
plotTernary(srt, features = gene, vertices = c("OS", "RE", "CH"))
```

`plotTernary.simMat` *Create quaternary simplex plot with pre-calculated similarity matrix*

Description

Create quaternary simplex plot with pre-calculated similarity matrix

Usage

```
## S3 method for class 'simMat'
plotTernary(
  x,
  title = NULL,
  veloMat = NULL,
  nGrid = 10,
  radius = 0.1,
  dotSize = NULL,
  dotShuffle = NULL,
  colorArg = NULL,
  labelColors = c("#3B4992FF", "#EE0000FF", "#008B45FF"),
  vertexLabelSize = NULL,
  vertexLabelDrift = 0.03,
  axisBreak = 5,
  axisTextShow = TRUE,
  axisTextSize = NULL,
  axisTextDrift = 0.01,
  gridLineAlpha = 0.6,
  arrowLinewidth = NULL,
  arrowAngle = 20,
  arrowLen = 0.2,
  titleSize = NULL,
  equilateral = TRUE,
  margin = 0.1,
  interactive = FALSE,
  ...
)
```

Arguments

`x` simMat object, n cells by 4 vertices, each row summing to 1.

`title` Title text of the plot. Default `NULL`.

`veloMat` Aggregated velocity matrix. Output of `aggrVeloGraph`.

`nGrid` Number of grids along the bottom side of the equilateral triangle. Default `10`.

`radius` Arrow length of unit velocity. Lower this when arrows point outside of the coordinate. Default `0.1`.

`dotSize` Dot aesthetics passed to `geom_point`. Default `0.6` when not interactive, `4` when interactive.

`dotShuffle` Whether to shuffle the order of dots being added to the plot, useful when categorical colors are used and mixing of categories is expected. Default `NULL` does shuffle when `dotColorBy` given is categorical and does not otherwise.

`colorArg` A "colorArg" object, internally prepared by `plotQuaternary.default`. Default `NULL`.

`labelColors` Colors of the axis lines and vertex labels. Default `c("#3B4992FF", "#EE0000FF", "#008B45FF")` (blue, red and green)

`vertexLabelSize`
Size of vertex labels. Default `6` when not interactive, `16` when interactive.

`vertexLabelDrift`
Position adjustment of the vertex labels, only applied to non-interactive view. Default `0.03`.

`axisBreak` Number of breaks to be labeled along axis. Default `5`.

axisTextShow Logical, whether to show axis text. Default TRUE.

axisTextSize Size of text along each axis break. Default 4 for non-interactive view, 12 for interactive view.

axisTextDrift Position adjustment of the axis text, only applied to non-interactive view. Default 0.01.

gridLineAlpha Transparency of background grid lines. Default 0.6.

arrowLinewidth Line width of the velocity arrows. Default 0.25 for non-interactive view, 2 for interactive view.

arrowAngle Controls the angle of the arrowhead, only applied to non-interactive view. Default 20.

arrowLen Control length in centimetre from arrow tip to arrow tail, only applied to non-interactive view. Default 0.2.

titleSize Size of title text. Default 14 for non-interactive view, 20 for interactive view.

equilateral Logical, whether to always display the triangle as equilateral. Default TRUE.

margin Margin allowed around of the triangle plotting region when equilateral = TRUE

interactive Logical. Whether to display plotly interactive view. Default FALSE.

... Not used

plotTernary.SingleCellExperiment

Create ternary simplex plot with SingleCellExperiment objects

Description

Create ternary plots that show similarity between single cells and selected three terminals in a ternary baricentric coordinate.

Usage

```
## S3 method for class 'SingleCellExperiment'
plotTernary(
  x,
  assay.type = "counts",
  clusterVar = NULL,
  processed = FALSE,
  dotColorBy = NULL,
  legendTitle = NULL,
  ...
)
```

Arguments

`x` A SingleCellExperiment object.

`assay.type` Assay to use for calculating the similarity. Default `"counts"`.

`clusterVar` A variable name in `colData(x)`. Default `NULL` uses `colLabels(x)`.

`processed` Logical. Whether the input matrix is already processed. `TRUE` will bypass internal preprocessing and input matrix will be directly used for similarity calculation. Default `FALSE` and raw count input is recommended. If missing in call, using `slot = "counts"` in "Seurat" method or using `assay.type = "counts"` in "SingleCellExperiment" method will force this argument to be `FALSE` and others for `TRUE`.

`dotColorBy` A vector/factor for coloring dots, can be either categorical (must be character or factor) or continuous. Default `NULL`.

`legendTitle` Title on the legend/colorbar. Default `NULL` uses `"cluster"` if `dotColorBy` is missing (default); user-end variable expression if `dotColorBy` is directly specified from plotQuaternary.default method; variable name if `dotColorBy` is specified from Seurat or SingleCellExperiment method.

`...` Arguments passed on to `plotTernary.default`, `plotTernary.simMat`

- `vertices` Vector of three unique cluster names that will be used for plotting. Or a named list that groups clusters as three terminal vertices. There must not be any overlap between groups.
- `features` Valid matrix row subsetting index to select features for similarity calculation. Default `NULL` uses all available features.
- `byCluster` Default `NULL` to generate one plot with all cells. Set `"all"` to split cells in plot by cluster and returns a list of subplots for each cluster as well as the plot including all cells. Otherwise, a vector of cluster names to generate a list of subplots for the specified clusters.
- `veloGraph` Cell x cell `dgCMatrix` object containing velocity information. Shows velocity grid-arrow layer when specified. Default `NULL` does not show velocity.
- `method` Similarity calculation method. Default `"euclidean"`. Choose from `"euclidean"`, `"cosine"`, `"pearson"`, `"spearman"`.
- `force` Whether to force calculate the similarity when more then 500 features are detected, which is generally not recommended. Default `FALSE`.
- `sigma` Gaussian kernel parameter that controls the effect of variance. Only effective when using a distance metric (i.e. `method` is `"euclidian"` or `"cosine"`). Larger values tighten the dot spreading on figure. Default `0.08`.
- `scale` Whether to min-max scale the distance matrix by clusters. Default `TRUE`.
- `dotColor` Character vector of color codes. When `dotColorBy` is `NULL`, use one or as many colors as the number of cells. If `dotColorBy` is categorical, specify as many colors as the number of categories in `dotColorBy` or ggplot2 categorical color palette is used by default. If `dotColorBy` is continuous, specify together with `breaks` argument.
- `palette` Color palette to use when `dotColorBy` is given. Default `"D"` (viridis) for continuous value and ggplot2 default for categorical value. See detail for alternatives.

`direction` Sets the order of colors in the scale. Default `1` orders as palette default. If `-1`, the order of colors is reversed.

`breaks` Number of breaks for continuous color scale passed to non-interactive "plot3D::scatter3D" call. Default `NULL`.

`returnData` Logical. Whether to return similarity and aggregated velocity data if applicable instead of generating plot. Default `FALSE`.

`title` Title text of the plot. Default `NULL`.

`nGrid` Number of grids along the bottom side of the equilateral triangle. Default `10`.

`radius` Arrow length of unit velocity. Lower this when arrows point outside of the coordinate. Default `0.1`.

`dotSize` Dot aesthetics passed to `geom_point`. Default `0.6` when not interactive, `4` when interactive.

`dotShuffle` Whether to shuffle the order of dots being added to the plot, useful when categorical colors are used and mixing of categories is expected. Default `NULL` does shuffle when `dotColorBy` given is categorical and does not otherwise.

`labelColors` Colors of the axis lines and vertex labels. Default `c("#3B4992FF", "#EE0000FF", "#008B45FF")` (blue, red and green)

`vertexLabelSize` Size of vertex labels. Default `6` when not interactive, `16` when interactive.

`vertexLabelDrift` Position adjustment of the vertex labels, only applied to non-interactive view. Default `0.03`.

`axisBreak` Number of breaks to be labeled along axis. Default `5`.

`axisTextShow` Logical, whether to show axis text. Default `TRUE`.

`axisTextSize` Size of text along each axis break. Default `4` for non-interactive view, `12` for interactive view.

`axisTextDrift` Position adjustment of the axis text, only applied to non-interactive view. Default `0.01`.

`gridLineAlpha` Transparency of background grid lines. Default `0.6`.

`arrowLinewidth` Line width of the velocity arrows. Default `0.25` for non-interactive view, `2` for interactive view.

`arrowAngle` Controls the angle of the arrowhead, only applied to non-interactive view. Default `20`.

`arrowLen` Control length in centimetre from arrow tip to arrow tail, only applied to non-interactive view. Default `0.2`.

`titleSize` Size of title text. Default `14` for non-interactive view, `20` for interactive view.

`equilateral` Logical, whether to always display the triangle as equilateral. Default `TRUE`.

`margin` Margin allowed around of the triangle plotting region when `equilateral = TRUE`

`interactive` Logical. Whether to display plotly interactive view. Default `FALSE`.

Value

By default, a "ggplot" object when `byCluster` is not specified, a list of "ggplot" object when `byCluster` is specified. When `interactive = TRUE`, a "plotly" object is returned. When `returnData = TRUE`, a list of similarity matrix and aggregated velocity matrix is returned.

See Also

Other plotTernary: `plotTernary()`, `plotTernary.Seurat()`

Examples

```
# SingleCellExperiment example
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(counts = rnaRaw))
colLabels(sce) <- rnaCluster
gene <- selectTopFeatures(sce, vertices = c("OS", "RE", "CH"))
plotTernary(sce, features = gene, vertices = c("OS", "RE", "CH"))
```

`print.quatPlot` *Show plist object produced with plot3D package*

Description

Show plist object produced with plot3D package

Usage

```
## S3 method for class 'quatPlot'
print(x, ...)
```

Arguments

`x`	quatPlot object, returned by `plotQuaternary` when `interactive = FALSE`.
`...`	Graphic parameters passed to `plot`. `mar` is pre-specified.

Value

No return value. It displays the plot described in a ’plist’ object returned by `plotQuaternary`, internally created by package ’plot3D’.

Examples

```
gene <- selectTopFeatures(
    x = rnaRaw,
    clusterVar = rnaCluster,
    vertices = c("RE", "OS", "CH", "ORT")
)
quat <- plotQuaternary(
    x = rnaRaw,
    clusterVar = rnaCluster,
    vertices = c("RE", "OS", "CH", "ORT"),
    features = gene,
    interactive = FALSE
)
quat; print(quat)
```

readH5ADObsNames *Extract 'adata.obs_names' from an H5AD file*

Description

It frequently happens that velocity analyses stored in H5AD files do not contain the full raw count data suggested for CytoSimplex visualization. Extracting the cell IDs (e.g. barcodes) helps matching the velocity data to raw count data imported from other sources.

Usage

```
readH5ADObsNames(filename)
```

Arguments

`filename`	File path to the H5AD file.

Value

A character vector of cell IDs.

See Also

Other H5AD-reader: `readH5ADObsVar()`, `readH5ADUnsSpMat()`, `readVelocytoLoom()`

Examples

```
## Not run:
h5adFile <- "path/to/analysis.h5ad"
cellIDs <- readH5ADObsNames(h5adFile)

## End(Not run)
```

readH5ADObsVar *Extract a variable from adata.obs stored in an H5AD file*

Description

Primarily designed for fetching the annotation used for visualization.

Usage

```
readH5ADObsVar(filename, obsKey, named = TRUE, categoricalAsFactor = TRUE)
```

Arguments

`filename` File path to the H5AD file.

`obsKey` The variable name to extract, must use only one character string.

`named` Logical, whether to name the vector with cell IDs that came from `adata.obs_names`. Default `TRUE`.

`categoricalAsFactor`
Logical, whether to convert categorical variables to factors. Default `TRUE`.

Value

A vector of the extracted variable, or a factor if the variable is encoded to be categorical and `categoricalAsFactor` = `TRUE`.

See Also

Other H5AD-reader: `readH5ADObsNames()`, `readH5ADUnsSpMat()`, `readVelocytoLoom()`

Examples

```
## Not run:
h5adFile <- "path/to/analysis.h5ad"
cluster <- readH5ADObsVar(h5adFile, "leiden")

## End(Not run)
```

readH5ADUnsSpMat *Extract a sparse matrix from adata.uns stored in an H5AD file*

Description

Primarily designed for fetching the velocity data presented as a cell-cell transition graph.

Usage

```
readH5ADUnsSpMat(filename, unsKey)
```

Arguments

`filename`	File path to the H5AD file.
`unsKey`	The `adata.uns` key to extract, must use only one character string.

Value

A CSC-matrix of "dgCMatrix" class

See Also

Other H5AD-reader: `readH5ADObsNames()`, `readH5ADObsVar()`, `readVelocytoLoom()`

Examples

```
## Not run:
h5adFile <- "path/to/analysis.h5ad"
velo <- readH5ADUnsSpMat(h5adFile, "velo_s_norm_graph")

## End(Not run)
```

readVelocytoLoom *Extract the raw counts from a LOOM file*

Description

This function is primarily designed for fetching the raw count data from a LOOM file, output by Velocyto. We by default use the spliced counts.

Usage

```
readVelocytoLoom(
  filename,
  matrixPath = "layers/spliced",
  cellID = "CellID",
  featureID = "Gene",
  chunkSize = 1000
)
```

Arguments

`filename`	File path to the LOOM file.
`matrixPath`	A path in the LOOM file to the matrix to extract, following the inner HDF5 structure. Default `"layers/spliced"`. See Details.
`cellID`	The name of the cell ID column in the LOOM column-attributes. The same thing as argument `obs_names` of `scanpy.read_loom`. Default `"CellID"`.
`featureID`	The name of the feature ID column in the LOOM row-attributes. The same thing as argument `var_names` of `scanpy.read_loom`. Default `"Gene"`.
`chunkSize`	The maximum size of the chunk to load the matrix. Default 1000.

Details

The velocyto output LOOM file is HDF5 based and is roughly organized as follows:

- `"matrix"`: The whole raw counts, which is the sum of spliced, unspliced and ambiguous counts.
- layers: A group like a folder
  - `"layers/spliced"`: The spliced counts.
  - `"layers/unspliced"`: The unspliced counts.
  - `"layers/ambiguous"`: The ambiguous counts.

An AnnData object created with Scanpy by default loads the data with a different structure, so that all the four matrices are accessible in `adata.layers` and set one of them (by default `"layers/spliced"`) to `adata.X`.

Value

A sparse matrix of class "dgCMatrix", with cells as columns and genes as rows.

See Also

Other H5AD-reader: `readH5ADObsNames()`, `readH5ADObsVar()`, `readH5ADUnsSpMat()`

Examples

```
## Not run:
loomFile <- "velocyto/out/analysis.loom"
rawCounts <- readVelocytoLoom(loomFile)

## End(Not run)
```

---

rnaCluster *Major cell type annotation of the example mouse bone marrow data*

Description

Major cell type annotation of the example mouse bone marrow data

Usage

```
rnaCluster
```

Format

`factor` object

Source

https://www.nature.com/articles/s41467-023-38034-2

References

Matsushita, Y., Liu, J., Chu, A.K.Y. et al. Bone marrow endosteal stem cells dictate active osteogenesis and aggressive tumorigenesis. Nat Commun 14, 2383 (2023).

---

rnaRaw *Mouse bone marrow scRNAseq example data*

Description

Mouse bone marrow scRNAseq example data

Usage

```
rnaRaw
```

Format

dgCMatrix-class object

Source

https://www.nature.com/articles/s41467-023-38034-2

References

Matsushita, Y., Liu, J., Chu, A.K.Y. et al. Bone marrow endosteal stem cells dictate active osteogenesis and aggressive tumorigenesis. Nat Commun 14, 2383 (2023).

`rnaVelo` *Velocity graph of the example mouse bone marrow data*

Description

Velocity graph of the example mouse bone marrow data

Usage

```
rnaVelo
```

Format

dgCMatrix-class object

Source

https://www.nature.com/articles/s41467-023-38034-2

References

Matsushita, Y., Liu, J., Chu, A.K.Y. et al. Bone marrow endosteal stem cells dictate active osteogenesis and aggressive tumorigenesis. Nat Commun 14, 2383 (2023).

`selectTopFeatures` *Pick top differentially presented features for similarity calculation*

Description

Performs wilcoxon rank-sum test on input matrix. While `clusterVar` and `vertices` together defines the groups of cells to be set as terminals of the simplex, this function will test each of these groups against the rest of the cells. The U-Statistics (`statistic`), p-value (`pval`) and adjusted p-value (`padj`), together with average presence in group (`avgExpr`), log fold-change (`logFC`), AUC (`auc`), percentage in group (`pct_in`) and percentage out of group (`pct_out`) will be calculated. Set `returnStats = TRUE` to return the full statistics table.

Top features are selected by sorting primarily on adjusted p-value, and secondarily on log fold-change, after filtering for up-regulated features.

Usage

```
selectTopFeatures(x, clusterVar, vertices, ...)

## Default S3 method:
selectTopFeatures(
  x,
  clusterVar,
  vertices,
  nTop = 30,
  processed = FALSE,
  lfcThresh = 0.1,
  returnStats = FALSE,
  ...
)

## S3 method for class 'Seurat'
selectTopFeatures(
  x,
  clusterVar = NULL,
  vertices,
  assay = NULL,
  layer = "counts",
  processed = FALSE,
  ...
)

## S3 method for class 'SingleCellExperiment'
selectTopFeatures(
  x,
  clusterVar = NULL,
  vertices,
  assay.type = "counts",
  processed = FALSE,
  ...
)
```

Arguments

`x` Dense or sparse matrix, observation per column. Preferrably a raw count matrix. Alternatively, a `Seurat` object or a `SingleCellExperiment` object.

`clusterVar` A vector/factor assigning the cluster variable to each column of the matrix object. For "Seurat" method, `NULL` (default) for `Idents(x)`, or a variable name in `meta.data` slot. For "SingleCellExperiment" method, `NULL` (default) for `colLabels(x)`, or a variable name in `colData` slot.

`vertices` Vector of cluster names that will be used for plotting. Or a named list that groups clusters as a terminal vertex. There must not be any overlap between groups.

`...` Arguments passed to methods.

`nTop`	Number of top differentially presented features per terminal. Default `30`.
`processed`	Logical. Whether the input matrix is already processed. `TRUE` will bypass internal preprocessing and input matrix will be directly used for rank-sum calculation. Default `FALSE` and raw count input is recommended.
`lfcThresh`	Threshold on log fold-change to identify up-regulated features. Default `0.1`.
`returnStats`	Logical. Whether to return the full statistics table rather then returning the selected genes. Default `FALSE`
`assay`	Assay name of the Seurat object to be used. Default `NULL`.
`layer`	For "Seurat" method, which layer of the assay to be used. Default `"counts"`.
`assay.type`	Assay name of the SingleCellExperiment object to be used. Default `"counts"`.

Value

When `returnStats` = `FALSE` (default), a character vector of at most `length(unique(vertices))*nTop` feature names. When `returnStats` = `TRUE`, a data.frame of wilcoxon rank sum test statistics.

Examples

```
selectTopFeatures(rnaRaw, rnaCluster, c("OS", "RE"))

# Seurat example
library(Seurat)
srt <- CreateSeuratObject(rnaRaw)
Idents(srt) <- rnaCluster
gene <- selectTopFeatures(srt, vertices = c("OS", "RE"))


# SingleCellExperiment example
library(SingleCellExperiment)
sce <- SingleCellExperiment(assays = list(counts = rnaRaw))
colLabels(sce) <- rnaCluster
gene <- selectTopFeatures(sce, vertices = c("OS", "RE"))
```

`writeQuaternaryGIF` *Create GIF image for dynamic rotating view of 3D quaternary simplex plot*

Description

Create GIF image for dynamic rotating view of 3D quaternary simplex plot

Usage

```
writeQuaternaryGIF(
  x,
  ...,
  cluster = NULL,
  filename = NULL,
  fps = 10,
  degreePerFrame = 10,
  width = 5,
  height = 5,
  res = 100
)
```

Arguments

`x` Input object that `plotQuaternary` accepts.

`...` All other arguments needed for `plotQuaternary`. Must be specified with exact argument names instead of a positional manner.

`cluster` One cluster that exists in `clusterVar` if users need to view the plot for a specific group. Default `NULL` plot all cells.

`filename` Output GIF image file path. Default `NULL` does not write to file.

`fps` Number of frame per second, must be a factor of 100. Default `10`.

`degreePerFrame` Number of degree that the tetrahedron is rotated per frame. Default `10`.

`width, height, res`
`grDevices::png` parameters to set figure size and resolutation. Width and Height are in inches. Default `5`, `5`, `100`.

Value

A object of class `magick-image` that can be shown in the Viewer panel in RStudio or equivalent display device. If `filename` is specified, the GIF image will be written to the file path.

Examples

```
gene <- selectTopFeatures(rnaRaw, rnaCluster, c("RE", "OS", "CH", "ORT"))

writeQuaternaryGIF(rnaRaw, clusterVar = rnaCluster, features = gene,
                   vertices = c("RE", "OS", "CH", "ORT"),
                   gifPath = tempfile(fileext = ".gif"))
```

Index

∗ **H5AD-reader**
readH5ADObsNames, 28
readH5ADObsVar, 29
readH5ADUnsSpMat, 30
readVelocytoLoom, 30
∗ **datasets**
rnaCluster, 32
rnaRaw, 32
rnaVelo, 33
∗ **plotQuaternary**
plotQuaternary, 6
plotQuaternary.Seurat, 9
plotQuaternary.SingleCellExperiment, 13
∗ **plotTernary**
plotTernary, 16
plotTernary.Seurat, 19
plotTernary.SingleCellExperiment, 24

arrow, *7*, *11*, *13*, *15*

colNormalize, 2

dgCMatrix-class, *32*, *33*

geom_point, *6*

plot, *27*
plotBinary, 4
plotQuaternary, 6, *12*, *16*, *27*, *36*
plotQuaternary.default, *10*, *13*, *14*, *23*
plotQuaternary.Seurat, *9*, 9, *16*
plotQuaternary.simMat, *7*, *10*, 12, *14*
plotQuaternary.SingleCellExperiment, *9*, *12*, 13
plotTernary, *4*, *6*, *9*, *13*, 16, *22*, *27*
plotTernary.default, *20*, *25*
plotTernary.Seurat, *19*, 19, *27*
plotTernary.simMat, *17*, *20*, 22, *25*
plotTernary.SingleCellExperiment, *19*, *22*, 24
png, *36*
print.quatPlot, 27

readH5ADObsNames, 28, *29–31*
readH5ADObsVar, *28*, 29, *30*, *31*
readH5ADUnsSpMat, *28*, *29*, 30, *31*
readVelocytoLoom, *28–30*, 30
rnaCluster, 32
rnaRaw, 32
rnaVelo, 33

selectTopFeatures, 33

writeQuaternaryGIF, *6*, *9*, *13*, 35