

Package: scDataviz (via r-universe)

June 6, 2026

Type Package

Title scDataviz: single cell dataviz and downstream analyses

Version 1.22.0

Description In the single cell World, which includes flow cytometry, mass cytometry, single-cell RNA-seq (scRNA-seq), and others, there is a need to improve data visualisation and to bring analysis capabilities to researchers even from non-technical backgrounds. scDataviz attempts to fit into this space, while also catering for advanced users. Additionally, due to the way that scDataviz is designed, which is based on SingleCellExperiment, it has a 'plug and play' feel, and immediately lends itself as flexible and compatible with studies that go beyond scDataviz. Finally, the graphics in scDataviz are generated via the ggplot engine, which means that users can 'add on' features to these with ease.

BugReports <https://github.com/kevinblighe/scDataviz/issues>

License GPL-3

Depends R (>= 4.0), S4Vectors, SingleCellExperiment,

Imports ggplot2, ggrepel, flowCore, umap, Seurat, reshape2, scales, RColorBrewer, corplot, stats, grDevices, graphics, utils, MASS, matrixStats, methods

Suggests PCAtools, cowplot, BiocGenerics, RUnit, knitr, kableExtra, rmarkdown

URL <https://github.com/kevinblighe/scDataviz>

biocViews SingleCell, ImmunoOncology, RNASeq, GeneExpression, Transcription, FlowCytometry, MassSpectrometry, DataImport

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.1.2

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev python3 zlib1g-dev

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

Date/Publication 2026-04-28 12:53:11 UTC

RemoteUrl <https://github.com/bioc/scDataviz>

RemoteRef RELEASE_3_23

RemoteSha 79f0dc43f3974f1f84a3a7290ddcfcf761349342

Contents

| | |
|--------------------------------------|-----------|
| basetheme | 2 |
| clusKNN | 4 |
| contourPlot | 6 |
| downsampleByVar | 9 |
| importData | 10 |
| markerEnrichment | 12 |
| markerExpression | 14 |
| markerExpressionPerCluster | 18 |
| metadataPlot | 20 |
| performUMAP | 24 |
| plotClusters | 25 |
| plotSignatures | 29 |
| processFCS | 30 |
| Index | 33 |

| | |
|-----------|--|
| basetheme | <i>Package-wide, non-user function used to set a base ggplot2 theme.</i> |
|-----------|--|

Description

Package-wide, non-user function used to set a base ggplot2 theme.

Usage

```
basetheme(
  titleLabSize,
  subtitleLabSize,
  captionLabSize,
  axisLabSize,
  xlabAngle,
  xlabhjust,
  xlabvjust,
  ylabAngle,
  ylabhjust,
  ylabvjust,
  legendPosition,
  legendLabSize
)
```

Arguments

| | |
|-----------------|--|
| titleLabSize | Size of plot title. |
| subtitleLabSize | Size of plot subtitle. |
| captionLabSize | Size of plot caption. |
| axisLabSize | Size of x- and y-axis labels. |
| xlabAngle | Rotation angle of x-axis labels. |
| xlabhjust | Horizontal adjustment of x-axis labels. |
| xlabvjust | Vertical adjustment of x-axis labels. |
| ylabAngle | Rotation angle of y-axis labels. |
| ylabhjust | Horizontal adjustment of y-axis labels. |
| ylabvjust | Vertical adjustment of y-axis labels. |
| legendPosition | Position of legend ('top', 'bottom', 'left', 'right', 'none'). |
| legendLabSize | Size of plot legend text. |

Details

Package-wide, non-user function used to set a base ggplot2 theme.

Value

A list object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create a theme
th <- basetheme(
  titleLabSize = 16,
  subtitleLabSize = 12,
  captionLabSize = 12,
  axisLabSize = 16,
  xlabAngle = 0,
  xlabhjust = 0.5,
  xlabvjust = 0.5,
  ylabAngle = 0,
  ylabhjust = 0.5,
  ylabvjust = 0.5,
  legendPosition = 'none',
  legendLabSize = 12)
```

`clusKNN`*A wrapper function for Seurat's FindNeighbors and FindClusters.*

Description

A wrapper function for Seurat's FindNeighbors and FindClusters.

Usage

```
clusKNN(  
  indata,  
  reducedDim = "UMAP",  
  dimColnames = c("UMAP1", "UMAP2"),  
  clusterAssignName = "Cluster",  
  distance.matrix = FALSE,  
  k.param = 20,  
  compute.SNN = TRUE,  
  prune.SNN = 1/15,  
  nn.method = "rann",  
  annoy.metric = "euclidean",  
  nn.eps = 0,  
  verbose = TRUE,  
  force.recalc = FALSE,  
  modularity.fxn = 1,  
  initial.membership = NULL,  
  weights = NULL,  
  node.sizes = NULL,  
  resolution = 0.8,  
  method = "matrix",  
  algorithm = 1,  
  n.start = 10,  
  n.iter = 10,  
  random.seed = 0,  
  group.singletons = TRUE,  
  temp.file.location = NULL,  
  edge.file.name = NULL,  
  overwrite = FALSE  
)
```

Arguments

| | |
|-------------------------|---|
| <code>indata</code> | A data-frame or matrix, or SingleCellExperiment object. If a SingleCellExperiment object, the cell-to-cluster assignments will be added as a new column, specified by <code>clusterAssignName</code> , to the input object's metadata; if a data-frame or matrix, only the cluster assignment vector is returned. |
| <code>reducedDim</code> | A reduced dimensional component stored within <code>indata</code> . e.g., PCA or UMAP. |

| | |
|---------------------------------|--|
| <code>dimColnames</code> | The column names of the dimensions to use. |
| <code>clusterAssignName</code> | The new column name in the metadata that will contain the determined cell-to-cluster assignments. |
| <code>distance.matrix</code> | Refer to <code>?Seurat::FindNeighbors</code> . |
| <code>k.param</code> | Refer to <code>?Seurat::FindNeighbors</code> . |
| <code>compute.SNN</code> | Refer to <code>?Seurat::FindNeighbors</code> . |
| <code>prune.SNN</code> | Refer to <code>?Seurat::FindNeighbors</code> . |
| <code>nn.method</code> | Refer to <code>?Seurat::FindNeighbors</code> . |
| <code>annoy.metric</code> | Refer to <code>?Seurat::FindNeighbors</code> . |
| <code>nn.eps</code> | Refer to <code>?Seurat::FindNeighbors</code> . |
| <code>verbose</code> | Refer to <code>?Seurat::FindNeighbors</code> . |
| <code>force.recalc</code> | Refer to <code>?Seurat::FindNeighbors</code> . |
| <code>modularity.fxn</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>initial.membership</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>weights</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>node.sizes</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>resolution</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>method</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>algorithm</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>n.start</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>n.iter</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>random.seed</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>group.singletons</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>temp.file.location</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>edge.file.name</code> | Refer to <code>?Seurat::FindClusters</code> . |
| <code>overwrite</code> | When the input object is a <code>SingleCellExperiment</code> , enabling this will result in the overwriting, with the new cluster assignments, of any column in your metadata that has the same name as <code>clusterAssignName</code> . |

Details

A wrapper function for Seurat's `FindNeighbors` and `FindClusters`.

Value

A `SingleCellExperiment` or numeric object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(1000), rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat) <- paste0('CD', 1:ncol(mat))
rownames(mat) <- paste0('cell', 1:nrow(mat))

clusKNN(mat)
```

| | |
|-------------|--|
| contourPlot | <i>Draw a contour plot, typically relating to co-ordinates of a 2-dimensional reduction / embedding, typically contained within a SingleCellExperiment object.</i> |
|-------------|--|

Description

Draw a contour plot, typically relating to co-ordinates of a 2-dimensional reduction / embedding, typically contained within a SingleCellExperiment object.

Usage

```
contourPlot(
  indata,
  reducedDim = "UMAP",
  dimColnames = c("UMAP1", "UMAP2"),
  lowcol = "darkblue",
  highcol = "darkred",
  alpha = c(0, 0.5),
  contour = "black",
  bins = 300,
  legendPosition = "right",
  legendLabSize = 12,
  legendIconSize = 5,
  legendKeyHeight = 2.5,
  xlim = NULL,
  ylim = NULL,
  celllab = NULL,
  labSize = 3,
  drawConnectors = TRUE,
  widthConnectors = 0.5,
  colConnectors = "black",
```

```

xlab = dimColnames[1],
xlabAngle = 0,
xlabhjust = 0.5,
xlabvjust = 0.5,
ylab = dimColnames[2],
ylabAngle = 0,
ylabhjust = 0.5,
ylabvjust = 0.5,
axisLabSize = 16,
title = "Cellular density and contours",
subtitle = "",
caption = ifelse(is(indata, "SingleCellExperiment"), paste0("Total cells, ",
  nrow(as.data.frame(reducedDim(indata, reducedDim))), "; Bins, ", bins),
  paste0("Total cells, ", nrow(indata), "; Bins, ", bins)),
titleLabSize = 16,
subtitleLabSize = 12,
captionLabSize = 12,
hline = NULL,
hlineType = "longdash",
hlineCol = "black",
hlineWidth = 0.4,
vline = NULL,
vlineType = "longdash",
vlineCol = "black",
vlineWidth = 0.4,
gridlines.major = TRUE,
gridlines.minor = TRUE,
borderWidth = 0.8,
borderColour = "black",
verbose = TRUE
)

```

Arguments

| | |
|-------------|--|
| indata | A data-frame or matrix, or SingleCellExperiment object. If a data-frame or matrix, columns named in dimColnames will be extracted from the data and used to generate the contour plot. If a SingleCellExperiment object, a reduction named by reducedDim will be taken from your object and used to generate the contour plot, again using columns whose names are specified in dimColnames. |
| reducedDim | A reduced dimensional embedding stored within indata, e.g., PCA or UMAP. |
| dimColnames | The column names of the dimensions to use. |
| lowcol | Shade for low-density contours. |
| highcol | Shade for high-density contours. |
| alpha | Control the gradient of colour transparency, with 1 being opaque. |
| contour | The colour of the contour lines. |
| bins | The number of bins that determine the overall density values. |

| | |
|-----------------|---|
| legendPosition | Position of legend ('top', 'bottom', 'left', 'right', 'none'). |
| legendLabSize | Size of plot legend text. |
| legendIconSize | Size of plot legend icons / symbols. |
| legendKeyHeight | Height of the legend key. |
| xlim | Limits of the x-axis. |
| ylim | Limits of the y-axis. |
| celllab | A vector containing any cells that the user wishes to label in the plot. |
| labSize | Size of labels. |
| drawConnectors | Logical, indicating whether or not to connect plot labels to their corresponding points by line connectors. |
| widthConnectors | Line width of connectors. |
| colConnectors | Line colour of connectors. |
| xlab | Label for x-axis. |
| xlabAngle | Rotation angle of x-axis labels. |
| xlabhjust | Horizontal adjustment of x-axis labels. |
| xlabvjust | Vertical adjustment of x-axis labels. |
| ylab | Label for y-axis. |
| ylabAngle | Rotation angle of y-axis labels. |
| ylabhjust | Horizontal adjustment of y-axis labels. |
| ylabvjust | Vertical adjustment of y-axis labels. |
| axisLabSize | Size of x- and y-axis labels. |
| title | Plot title. |
| subtitle | Plot subtitle. |
| caption | Plot caption. |
| titleLabSize | Size of plot title. |
| subtitleLabSize | Size of plot subtitle. |
| captionLabSize | Size of plot caption. |
| hline | Draw one or more horizontal lines passing through this/these values on y-axis. For single values, only a single numerical value is necessary. For multiple lines, pass these as a vector, e.g., c(60,90). |
| hlineType | Line type for hline ('blank', 'solid', 'dashed', 'dotted', 'dotdash', 'longdash', 'twodash'). |
| hlineCol | Colour of hline. |
| hlineWidth | Width of hline. |
| vline | Draw one or more vertical lines passing through this/these values on x-axis. For single values, only a single numerical value is necessary. For multiple lines, pass these as a vector, e.g., c(60,90). |

| | |
|-----------------|---|
| vlineType | Line type for vline ('blank', 'solid', 'dashed', 'dotted', 'dotdash', 'longdash', 'twodash'). |
| vlineCol | Colour of vline. |
| vlineWidth | Width of vline. |
| gridlines.major | Logical, indicating whether or not to draw major gridlines. |
| gridlines.minor | Logical, indicating whether or not to draw minor gridlines. |
| borderWidth | Width of the border on the x and y axes. |
| borderColour | Colour of the border on the x and y axes. |
| verbose | Boolean (TRUE / FALSE) to print messages to console or not. |

Details

Draw a contour plot, typically relating to co-ordinates of a 2-dimensional reduction / embedding, typically contained within a SingleCellExperiment object.

Value

A ggplot2 object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(1000, rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat) <- paste0('CD', 1:ncol(mat))

u <- umap::umap(mat)$layout
colnames(u) <- c('UMAP1', 'UMAP2')

contourPlot(u)
```

downsampleByVar

Downsample an input data-frame or matrix based on variance.

Description

Downsample an input data-frame or matrix based on variance.

Usage

```
downsampleByVar(x, varianceFactor = 0.1, verbose = TRUE)
```

Arguments

x Input data-matrix.
varianceFactor Removes this proportion of variables based on lesser variance.
verbose Boolean (TRUE / FALSE) to print messages to console or not.

Details

Downsample an input data-frame or matrix based on variance.

Value

A matrix object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(1000, rate=.1), theta = 4.5),
  ncol = 20))

downsampleByVar(mat, varianceFactor = 0.1)
```

importData *Import a data-frame or matrix, and associated metadata, to a SingleCellExperiment object.*

Description

Import a data-frame or matrix, and associated metadata, to a SingleCellExperiment object.

Usage

```
importData(
  mat,
  assayname,
  metadata = NULL,
  downsampleVar = NULL,
  verbose = TRUE
)
```

Arguments

| | |
|---------------|---|
| mat | A data-frame or matrix of expression values. Data-frames will be coerced to matrices. |
| assayname | Name of the SingleCellExperiment assay slot in which data will be stored. |
| metadata | Metadata associated with the data contained in 'mat'. A strict rule is enforced requiring that <code>rownames(metadata) == rownames(mat)</code> . |
| downsampleVar | Downsample based on variance. Removes this proportion of variables (rows) based on lesser variance. This is applied on a per sample basis. If user wishes to apply this globally on the final merged dataset, then set this to 0 and remove based on variance manually after object creation. |
| verbose | Boolean (TRUE / FALSE) to print messages to console or not. |

Details

Import a data-frame or matrix, and associated metadata, to a SingleCellExperiment object.

Value

A SingleCellExperiment object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(50000), rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat) <- paste0('CD', 1:ncol(mat))
rownames(mat) <- paste0('cell', 1:nrow(mat))

metadata <- data.frame(
  group = rep('A', nrow(mat)),
  row.names = rownames(mat),
  stringsAsFactors = FALSE)

sce <- importData(mat,
  assayname = 'normcounts',
  metadata = metadata)
```

| | |
|------------------|---|
| markerEnrichment | <i>Find enriched markers per identified cluster and calculate cluster abundances across these for samples and metadata variables.</i> |
|------------------|---|

Description

Find enriched markers per identified cluster and calculate cluster abundances across these for samples and metadata variables.

Usage

```
markerEnrichment(
  indata,
  meta = NULL,
  assay = "scaled",
  sampleAbundances = TRUE,
  sampleID = "sample",
  studyvarID = NULL,
  clusterAssign = metadata(indata)[["Cluster"]],
  funcSummarise = function(x) mean(x, na.rm = TRUE),
  method = "Z",
  prob = 0.1,
  limits = c(-1.96, 1.96),
  verbose = TRUE
)
```

Arguments

| | |
|------------------|--|
| indata | A data-frame or matrix, or SingleCellExperiment object. If a data-frame or matrix, this should relate to expression data (cells as columns; genes as rows). If a SingleCellExperiment object, data will be extracted from an assay component named by assay. |
| meta | If 'indata' is a non-SingleCellExperiment object, meta must be activated and relate to a data-frame of metadata that aligns with the columns of indata, and that also contains a column name specified by studyvarID. |
| assay | Name of the assay slot in indata from which data will be taken, assuming indata is a SingleCellExperiment object. |
| sampleAbundances | Logical, indicating whether or not to calculate cluster abundances across study samples. |
| sampleID | If sampleAbundances == TRUE, a column name from the provided metadata representing over which sample cluster abundances will be calculated. |
| studyvarID | A column name from the provided metadata representing a condition or trait over which cluster abundances will be calculated. |

| | |
|---------------|--|
| clusterAssign | A vector of cell-to-cluster assignments. This can be from any source but must align with your cells / variables. There is no check to ensure this when 'indata' is not a SingleCellExperiment object. |
| funcSummarise | A mathematical function used to summarise expression per marker per cluster. |
| method | Type of summarisation to apply to the data for final marker selection. Possible values include Z or quantile. If Z, limits relate to lower and upper Z-score cut-offs for lowhigh markers. The defaults of -1.96 and +1.96 are equivalents of $p < 0.05$ on a two-tailed distribution. If quantile, prob will be used to define the nth lower and 1 - nth upper quantiles, which will be used for selecting lowhigh markers. |
| prob | See details for method. |
| limits | See details for method. |
| verbose | Boolean (TRUE / FALSE) to print messages to console or not. |

Details

Find enriched markers per identified cluster and calculate cluster abundances across these for samples and metadata variables. markerEnrichment first collapses your input data's expression profiles from the level of cells to the level of clusters based on a mathematical function specified by funcSummarise. It then either selects, per cluster, lowhigh markers via quantiles, or transforms this collapsed data to global Z-scores and selects lowhigh markers based on Z-score cut-offs.

Value

A data.frame object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(1000, rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat) <- paste0('CD', 1:ncol(mat))
rownames(mat) <- paste0('cell', 1:nrow(mat))

u <- umap::umap(mat)$layout
colnames(u) <- c('UMAP1', 'UMAP2')
rownames(u) <- rownames(mat)
clus <- clusKNN(u)

metadata <- data.frame(
  group = c(rep('PB1', 25), rep('PB2', 25)),
  row.names = rownames(u))

markerEnrichment(t(mat), meta = metadata,
```

```
sampleAbundances = FALSE,
studyvarID = 'group', clusterAssign = clus)
```

| | |
|------------------|---|
| markerExpression | <i>Highlight the individual marker expression profile across a 2-dimensional reduction / embedding, typically contained within a SingleCellExperiment object. By default, this function plots the expression profile of 6 randomly-selected markers from your data.</i> |
|------------------|---|

Description

Highlight the individual marker expression profile across a 2-dimensional reduction / embedding, typically contained within a SingleCellExperiment object. By default, this function plots the expression profile of 6 randomly-selected markers from your data.

Usage

```
markerExpression(
  indata,
  layout = NULL,
  assay = "scaled",
  reducedDim = "UMAP",
  dimColnames = c("UMAP1", "UMAP2"),
  markers = sample(rownames(indata), 6),
  ncol = 3,
  nrow = 2,
  col = c("darkblue", "yellow"),
  colMidpoint = 0,
  alpha = c(0, 1),
  pointSize = 0.5,
  legendPosition = "right",
  legendLabSize = 12,
  legendKeyHeight = 2.5,
  xlim = NULL,
  ylim = NULL,
  celllab = NULL,
  labSize = 3,
  drawConnectors = TRUE,
  widthConnectors = 0.5,
  colConnectors = "black",
  xlab = dimColnames[1],
  xlabAngle = 0,
  xlabhjust = 0.5,
  xlabvjust = 0.5,
  ylab = dimColnames[2],
  ylabAngle = 0,
```

```

ylabhjust = 0.5,
ylabvjust = 0.5,
axisLabSize = 16,
stripLabSize = 16,
title = "Individual marker expression",
subtitle = "",
caption = ifelse(is(indata, "SingleCellExperiment"), paste0("Total cells, ",
  nrow(as.data.frame(reducedDim(indata, reducedDim))))), paste0("Total cells, ",
  nrow(layout))),
titleLabSize = 16,
subtitleLabSize = 12,
captionLabSize = 12,
hline = NULL,
hlineType = "longdash",
hlineCol = "black",
hlineWidth = 0.4,
vline = NULL,
vlineType = "longdash",
vlineCol = "black",
vlineWidth = 0.4,
gridlines.major = TRUE,
gridlines.minor = TRUE,
borderWidth = 0.8,
borderColour = "black"
)

```

Arguments

| | |
|-------------|--|
| indata | A data-frame or matrix, or SingleCellExperiment object. If a data-frame or matrix, this should relate to expression data (cells as columns; genes as rows). If a SingleCellExperiment object, data will be extracted from an assay component named by assay. |
| layout | If 'indata' is a non-SingleCellExperiment object, layout must be activated and relate to a 2-dimensional reduction / embedding, although, technically, any data-frame or matrix of numbers will be accepted, provided that it aligns with the dimensions of indata, and provided that it contains columns as specified in dimColnames. |
| assay | Name of the assay slot in 'indata' from which data will be taken, assuming indata is a SingleCellExperiment object. |
| reducedDim | A reduced dimensional component stored within indata, e.g., PCA or UMAP. |
| dimColnames | The column names of the dimensions to use. |
| markers | Vector containing marker names to plot. |
| ncol | Number of columns for faceting. |
| nrow | Number of rows for faceting. |
| col | Colours used for generation of fill gradient according to expression values. Can be 2 or 3 colours. |

| | |
|------------------------------|---|
| <code>colMidpoint</code> | Mid-point (expression value) for the colour range. Only used when 3 colours are specified by <code>col</code> . |
| <code>alpha</code> | Control the gradient of colour transparency, with 1 being opaque. |
| <code>pointSize</code> | Size of plotted points. |
| <code>legendPosition</code> | Position of legend ('top', 'bottom', 'left', 'right', 'none'). |
| <code>legendLabSize</code> | Size of plot legend text. |
| <code>legendKeyHeight</code> | Height of the legend key. |
| <code>xlim</code> | Limits of the x-axis. |
| <code>ylim</code> | Limits of the y-axis. |
| <code>celllab</code> | A vector containing any cells that the user wishes to label in the plot. |
| <code>labSize</code> | Size of labels. |
| <code>drawConnectors</code> | Logical, indicating whether or not to connect plot labels to their corresponding points by line connectors. |
| <code>widthConnectors</code> | Line width of connectors. |
| <code>colConnectors</code> | Line colour of connectors. |
| <code>xlab</code> | Label for x-axis. |
| <code>xlabAngle</code> | Rotation angle of x-axis labels. |
| <code>xlabhjust</code> | Horizontal adjustment of x-axis labels. |
| <code>xlabvjust</code> | Vertical adjustment of x-axis labels. |
| <code>ylab</code> | Label for y-axis. |
| <code>ylabAngle</code> | Rotation angle of y-axis labels. |
| <code>ylabhjust</code> | Horizontal adjustment of y-axis labels. |
| <code>ylabvjust</code> | Vertical adjustment of y-axis labels. |
| <code>axisLabSize</code> | Size of x- and y-axis labels. |
| <code>stripLabSize</code> | Size of the strip (marker) labels. |
| <code>title</code> | Plot title. |
| <code>subtitle</code> | Plot subtitle. |
| <code>caption</code> | Plot caption. |
| <code>titleLabSize</code> | Size of plot title. |
| <code>subtitleLabSize</code> | Size of plot subtitle. |
| <code>captionLabSize</code> | Size of plot caption. |
| <code>hline</code> | Draw one or more horizontal lines passing through this/these values on y-axis. For single values, only a single numerical value is necessary. For multiple lines, pass these as a vector, e.g., <code>c(60,90)</code> . |
| <code>hlineType</code> | Line type for <code>hline</code> ('blank', 'solid', 'dashed', 'dotted', 'dotdash', 'longdash', 'twodash'). |

| | |
|-----------------|---|
| hlineCol | Colour of hline. |
| hlineWidth | Width of hline. |
| vline | Draw one or more vertical lines passing through this/these values on x-axis. For single values, only a single numerical value is necessary. For multiple lines, pass these as a vector, e.g., c(60,90). |
| vlineType | Line type for vline ('blank', 'solid', 'dashed', 'dotted', 'dotdash', 'longdash', 'twodash'). |
| vlineCol | Colour of vline. |
| vlineWidth | Width of vline. |
| gridlines.major | Logical, indicating whether or not to draw major gridlines. |
| gridlines.minor | Logical, indicating whether or not to draw minor gridlines. |
| borderWidth | Width of the border on the x and y axes. |
| borderColour | Colour of the border on the x and y axes. |

Details

Highlight the individual marker expression profile across a 2-dimensional reduction / embedding, typically contained within a `SingleCellExperiment` object. By default, this function plots the expression profile of 6 randomly-selected markers from your data.

Value

A `ggplot2` object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(1000, rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat) <- paste0('CD', 1:ncol(mat))
rownames(mat) <- paste0('cell', 1:nrow(mat))

u <- umap::umap(mat)$layout
colnames(u) <- c('UMAP1', 'UMAP2')
rownames(u) <- rownames(mat)

markerExpression(t(mat), layout = u)
```

markerExpressionPerCluster

Generate box-and-whisker plots illustrating marker expression per k-NN identified cluster. By default, 5 randomly-selected clusters are selected, and the expression profiles of 10 randomly-selected markers are plot across these.

Description

Generate box-and-whisker plots illustrating marker expression per k-NN identified cluster. By default, 5 randomly-selected clusters are selected, and the expression profiles of 10 randomly-selected markers are plot across these.

Usage

```
markerExpressionPerCluster(
  indata,
  assay = "scaled",
  clusters = sample(unique(metadata(indata)[["Cluster"]]), 5),
  clusterAssign = metadata(indata)[["Cluster"]],
  markers = sample(rownames(indata), 10),
  ncol = 5,
  nrow = 2,
  legendPosition = "none",
  legendLabSize = 12,
  legendKeyHeight = 2.5,
  xlim = NULL,
  ylim = NULL,
  yfixed = FALSE,
  xlab = "Marker",
  xlabAngle = 90,
  xlabhjust = 0.5,
  xlabvjust = 0.5,
  ylab = "Expression",
  ylabAngle = 0,
  ylabhjust = 0.5,
  ylabvjust = 0.5,
  axisLabSize = 16,
  stripLabSize = 16,
  title = "Marker expression per cluster",
  subtitle = "",
  caption = "",
  titleLabSize = 16,
  subtitleLabSize = 12,
  captionLabSize = 12,
  borderWidth = 0.8,
  borderColour = "black",
```

```

    verbose = TRUE
  )

```

Arguments

| | |
|-----------------|--|
| indata | A data-frame or matrix, or SingleCellExperiment object. If a data-frame or matrix, this should relate to expression data (cells as columns; genes as rows). If a SingleCellExperiment object, data will be extracted from an assay component named by assay. |
| assay | Name of the assay slot in indata from which data will be taken, assuming indata is a SingleCellExperiment object. |
| clusters | Vector containing clusters to plot. |
| clusterAssign | A vector of cell-to-cluster assignments. This can be from any source but must align with your cells / variables. There is no check to ensure this when indata is not a SingleCellExperiment object. |
| markers | Vector containing marker names to plot. |
| ncol | Number of columns for faceting. |
| nrow | Number of rows for faceting. |
| legendPosition | Position of legend ('top', 'bottom', 'left', 'right', 'none'). |
| legendLabSize | Size of plot legend text. |
| legendKeyHeight | Height of the legend key. |
| xlim | Limits of the x-axis. |
| ylim | Limits of the y-axis. |
| yfixed | Logical, specifying whether or not to fix the y-axis scales across all clusters when faceting. |
| xlab | Label for x-axis. |
| xlabAngle | Rotation angle of x-axis labels. |
| xlabhjust | Horizontal adjustment of x-axis labels. |
| xlabvjust | Vertical adjustment of x-axis labels. |
| ylab | Label for y-axis. |
| ylabAngle | Rotation angle of y-axis labels. |
| ylabhjust | Horizontal adjustment of y-axis labels. |
| ylabvjust | Vertical adjustment of y-axis labels. |
| axisLabSize | Size of x- and y-axis labels. |
| stripLabSize | Size of the strip labels. |
| title | Plot title. |
| subtitle | Plot subtitle. |
| caption | Plot caption. |
| titleLabSize | Size of plot title. |

| | |
|-----------------|---|
| subtitleLabSize | Size of plot subtitle. |
| captionLabSize | Size of plot caption. |
| borderWidth | Width of the border on the x and y axes. |
| borderColour | Colour of the border on the x and y axes. |
| verbose | Boolean (TRUE / FALSE) to print messages to console or not. |

Details

Generate box-and-whisker plots illustrating marker expression per k-NN identified cluster. By default, 5 randomly-selected clusters are selected, and the expression profiles of 10 randomly-selected markers are plot across these.

Value

A ggplot2 object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(5000, rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat) <- paste0('CD', 1:ncol(mat))
rownames(mat) <- paste0('cell', 1:nrow(mat))

clus <- clusKNN(mat)
markerExpressionPerCluster(t(mat), clusters = c(0, 1),
  clusterAssign = clus)
```

metadataPlot

Colour shade a 2-dimensional reduction / embedding based on metadata, typically contained within a SingleCellExperiment object.

Description

Colour shade a 2-dimensional reduction / embedding based on metadata, typically contained within a SingleCellExperiment object.

Usage

```
metadataPlot(  
  indata,  
  meta = NULL,  
  reducedDim = "UMAP",  
  dimColnames = c("UMAP1", "UMAP2"),  
  colby = NULL,  
  colkey = NULL,  
  pointSize = 0.5,  
  legendPosition = "right",  
  legendLabSize = 12,  
  legendIconSize = 5,  
  xlim = NULL,  
  ylim = NULL,  
  celllab = NULL,  
  labSize = 3,  
  drawConnectors = TRUE,  
  widthConnectors = 0.5,  
  colConnectors = "black",  
  xlab = dimColnames[1],  
  xlabAngle = 0,  
  xlabhjust = 0.5,  
  xlabvjust = 0.5,  
  ylab = dimColnames[2],  
  ylabAngle = 0,  
  ylabhjust = 0.5,  
  ylabvjust = 0.5,  
  axisLabSize = 16,  
  title = "Metadata plot",  
  subtitle = "",  
  caption = ifelse(is(indata, "SingleCellExperiment"), paste0("Total cells, ",  
    nrow(as.data.frame(reducedDim(indata, reducedDim)))), paste0("Total cells, ",  
    nrow(meta))),  
  titleLabSize = 16,  
  subtitleLabSize = 12,  
  captionLabSize = 12,  
  hline = NULL,  
  hlineType = "longdash",  
  hlineCol = "black",  
  hlineWidth = 0.4,  
  vline = NULL,  
  vlineType = "longdash",  
  vlineCol = "black",  
  vlineWidth = 0.4,  
  gridlines.major = TRUE,  
  gridlines.minor = TRUE,  
  borderWidth = 0.8,  
  borderColour = "black"
```

)

Arguments

| | |
|-----------------|--|
| indata | A data-frame or matrix, or SingleCellExperiment object. If a data-frame or matrix, columns named in dimColnames will be extracted from the data and used to generate the plot. If a SingleCellExperiment object, a reduction named by reducedDim will be taken from your object and used to generate the plot, again using columns whose names are specified in dimColnames. |
| meta | If 'indata' is a non-SingleCellExperiment object, 'meta' must be activated and relate to a data-frame of metadata that aligns with the rows of indata, and that also contains a column name specified by colby. |
| reducedDim | A reduced dimensional embedding stored within indata, e.g., PCA or UMAP. |
| dimColnames | The column names of the dimensions to use. |
| colby | If NULL, all points will be coloured differently. If not NULL, the value is assumed to be a column name in metadata(indata) relating to some grouping / categorical variable. |
| colkey | Vector of name-value pairs relating to value passed to 'col', e.g., c(A='forestgreen', B='gold'). |
| pointSize | Size of plotted points. |
| legendPosition | Position of legend ('top', 'bottom', 'left', 'right', 'none'). |
| legendLabSize | Size of plot legend text. |
| legendIconSize | Size of plot legend icons / symbols. |
| xlim | Limits of the x-axis. |
| ylim | Limits of the y-axis. |
| celllab | A vector containing any cells that the user wishes to label in the plot. |
| labSize | Size of labels. |
| drawConnectors | Logical, indicating whether or not to connect plot labels to their corresponding points by line connectors. |
| widthConnectors | Line width of connectors. |
| colConnectors | Line colour of connectors. |
| xlab | Label for x-axis. |
| xlabAngle | Rotation angle of x-axis labels. |
| xlabhjust | Horizontal adjustment of x-axis labels. |
| xlabvjust | Vertical adjustment of x-axis labels. |
| ylab | Label for y-axis. |
| ylabAngle | Rotation angle of y-axis labels. |
| ylabhjust | Horizontal adjustment of y-axis labels. |
| ylabvjust | Vertical adjustment of y-axis labels. |
| axisLabSize | Size of x- and y-axis labels. |

| | |
|-----------------|---|
| title | Plot title. |
| subtitle | Plot subtitle. |
| caption | Plot caption. |
| titleLabSize | Size of plot title. |
| subtitleLabSize | Size of plot subtitle. |
| captionLabSize | Size of plot caption. |
| hline | Draw one or more horizontal lines passing through this/these values on y-axis. For single values, only a single numerical value is necessary. For multiple lines, pass these as a vector, e.g., c(60,90). |
| hlineType | Line type for hline ('blank', 'solid', 'dashed', 'dotted', 'dotdash', 'longdash', 'twodash'). |
| hlineCol | Colour of hline. |
| hlineWidth | Width of hline. |
| vline | Draw one or more vertical lines passing through this/these values on x-axis. For single values, only a single numerical value is necessary. For multiple lines, pass these as a vector, e.g., c(60,90). |
| vlineType | Line type for vline ('blank', 'solid', 'dashed', 'dotted', 'dotdash', 'longdash', 'twodash'). |
| vlineCol | Colour of vline. |
| vlineWidth | Width of vline. |
| gridlines.major | Logical, indicating whether or not to draw major gridlines. |
| gridlines.minor | Logical, indicating whether or not to draw minor gridlines. |
| borderWidth | Width of the border on the x and y axes. |
| borderColour | Colour of the border on the x and y axes. |

Details

Colour shade a 2-dimensional reduction / embedding based on metadata, typically contained within a SingleCellExperiment object.

Value

A ggplot2 object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(1000), rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat) <- paste0('CD', 1:ncol(mat))
rownames(mat) <- paste0('cell', 1:nrow(mat))

u <- umap::umap(mat)$layout
colnames(u) <- c('UMAP1', 'UMAP2')
rownames(u) <- rownames(mat)

metadata <- data.frame(
  group = c(rep('PB1', 25), rep('PB2', 25)),
  row.names = rownames(u))

metadataPlot(u, meta = metadata, colby = 'group')
```

| | |
|-------------|---|
| performUMAP | <i>Perform UMAP on an input data-frame or matrix, or SingleCellExperiment object, using the basic R implementation of UMAP.</i> |
|-------------|---|

Description

Perform UMAP on an input data-frame or matrix, or SingleCellExperiment object, using the basic R implementation of UMAP.

Usage

```
performUMAP(
  indata,
  config = NULL,
  assay = "scaled",
  reducedDim = NULL,
  dims = seq_len(20),
  newDimName = NULL,
  useMarkers = NULL,
  verbose = TRUE
)
```

Arguments

| | |
|--------|---|
| indata | A data-frame or matrix, or SingleCellExperiment object. If a data-frame or matrix, only the derived co-ordinates for the first two dimensions are returned. If a SingleCellExperiment object, UMAP is performed on the assay named by assay, and the co-ordinates for the first two dimensions are stored as a reduced dimension named by reducedDim. |
|--------|---|

| | |
|------------|---|
| config | UMAP configuration settings |
| assay | Name of the assay slot in <code>indata</code> from which data will be taken, assuming <code>indata</code> is a <code>SingleCellExperiment</code> object. |
| reducedDim | A dimensional reduction / embedding stored within <code>indata</code> , e.g., PCA. If activated, UMAP will be performed on this object in place of the assay component specified by <code>assay</code> . |
| dims | If 'reducedDim' is activated, the number of dimensions to use. |
| newDimName | Name for the new dimensional embedding that will be produced. If nothing is selected for neither this nor <code>reducedDim</code> , then the new name will be UMAP. If nothing is selected for this, but, e.g., PCA is selected for <code>reducedDim</code> , then the new name will be UMAP_PCA. |
| useMarkers | Before performing UMAP, subset the data for these markers. |
| verbose | Boolean (TRUE / FALSE) to print messages to console or not. |

Details

Perform UMAP on an input data-frame or matrix, or `SingleCellExperiment` object, using the basic R implementation of UMAP.

Value

A `SingleCellExperiment` object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(1000, rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat) <- paste0('CD', 1:ncol(mat))

performUMAP(mat)
```

| | |
|--------------|--|
| plotClusters | <i>Highlight cell-to-cluster assignments across a 2-dimensional reduction / embedding.</i> |
|--------------|--|

Description

Highlight cell-to-cluster assignments across a 2-dimensional reduction / embedding.

Usage

```
plotClusters(  
  indata,  
  clusterVector = NULL,  
  reducedDim = "UMAP",  
  dimColnames = c("UMAP1", "UMAP2"),  
  clusterColname = "Cluster",  
  pointSize = 0.5,  
  legendPosition = "none",  
  legendLabSize = 12,  
  xlim = NULL,  
  ylim = NULL,  
  label = TRUE,  
  labSize = 5,  
  drawConnectors = TRUE,  
  widthConnectors = 0.5,  
  colConnectors = "black",  
  xlab = dimColnames[1],  
  xlabAngle = 0,  
  xlabhjust = 0.5,  
  xlabvjust = 0.5,  
  ylab = dimColnames[2],  
  ylabAngle = 0,  
  ylabhjust = 0.5,  
  ylabvjust = 0.5,  
  axisLabSize = 16,  
  title = "k-nearest neighbour (k-NN) clusters",  
  subtitle = "",  
  caption = ifelse(is(indata, "SingleCellExperiment"), paste0("Total cells, ",  
    nrow(as.data.frame(reducedDim(indata, reducedDim)))), paste0("Total cells, ",  
    length(clusterVector))),  
  titleLabSize = 16,  
  subtitleLabSize = 12,  
  captionLabSize = 12,  
  hline = NULL,  
  hlineType = "longdash",  
  hlineCol = "black",  
  hlineWidth = 0.4,  
  vline = NULL,  
  vlineType = "longdash",  
  vlineCol = "black",  
  vlineWidth = 0.4,  
  gridlines.major = TRUE,  
  gridlines.minor = TRUE,  
  borderWidth = 0.8,  
  borderColour = "black",  
  verbose = TRUE  
)
```

Arguments

| | |
|-----------------|--|
| indata | A data-frame or matrix, or SingleCellExperiment object. If a data-frame or matrix, columns named in dimColnames will be extracted from the data and used to generate the plot. If a SingleCellExperiment object, a reduction named by reducedDim will be taken from your object and used to generate the plot, again using columns whose names are specified in dimColnames. |
| clusterVector | If indata is a non-SingleCellExperiment object, clusterVector must be non-NULL and relate to a cell-to-cluster assignment whose length matches nrow(indata). |
| reducedDim | A reduced dimensional embedding stored within 'indata', e.g., PCA or UMAP. |
| dimColnames | The column names of the dimensions to use. |
| clusterColname | The column name in the metadata of indata that contains the cell-to-cluster assignment, assuming indata is a SingleCellExperiment object. |
| pointSize | Size of plotted points. |
| legendPosition | Position of legend ('top', 'bottom', 'left', 'right', 'none'). |
| legendLabSize | Size of plot legend text. |
| xlim | Limits of the x-axis. |
| ylim | Limits of the y-axis. |
| label | Logical, indicating whether or not to label the clusters. |
| labSize | Size of labels. |
| drawConnectors | Logical, indicating whether or not to connect plot labels to their corresponding cluster islands by line connectors. |
| widthConnectors | Line width of connectors. |
| colConnectors | Line colour of connectors. |
| xlab | Label for x-axis. |
| xlabAngle | Rotation angle of x-axis labels. |
| xlabhjust | Horizontal adjustment of x-axis labels. |
| xlabvjust | Vertical adjustment of x-axis labels. |
| ylab | Label for y-axis. |
| ylabAngle | Rotation angle of y-axis labels. |
| ylabhjust | Horizontal adjustment of y-axis labels. |
| ylabvjust | Vertical adjustment of y-axis labels. |
| axisLabSize | Size of x- and y-axis labels. |
| title | Plot title. |
| subtitle | Plot subtitle. |
| caption | Plot caption. |
| titleLabSize | Size of plot title. |
| subtitleLabSize | Size of plot subtitle. |

| | |
|-----------------|---|
| captionLabSize | Size of plot caption. |
| hline | Draw one or more horizontal lines passing through this/these values on y-axis. For single values, only a single numerical value is necessary. For multiple lines, pass these as a vector, e.g., c(60,90). |
| hlineType | Line type for hline ('blank', 'solid', 'dashed', 'dotted', 'dotdash', 'longdash', 'twodash'). |
| hlineCol | Colour of hline. |
| hlineWidth | Width of hline. |
| vline | Draw one or more vertical lines passing through this/these values on x-axis. For single values, only a single numerical value is necessary. For multiple lines, pass these as a vector, e.g., c(60,90). |
| vlineType | Line type for vline ('blank', 'solid', 'dashed', 'dotted', 'dotdash', 'longdash', 'twodash'). |
| vlineCol | Colour of vline. |
| vlineWidth | Width of vline. |
| gridlines.major | Logical, indicating whether or not to draw major gridlines. |
| gridlines.minor | Logical, indicating whether or not to draw minor gridlines. |
| borderWidth | Width of the border on the x and y axes. |
| borderColour | Colour of the border on the x and y axes. |
| verbose | Boolean (TRUE / FALSE) to print messages to console or not. |

Details

Highlight cell-to-cluster assignments across a 2-dimensional reduction / embedding.

Value

A ggplot2 object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(1000, rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat) <- paste0('CD', 1:ncol(mat))
rownames(mat) <- paste0('cell', 1:nrow(mat))

u <- umap::umap(mat)
clusvec <- clusKNN(u$layout)
plotClusters(u$layout, clusvec)
```

| | |
|----------------|--|
| plotSignatures | <i>Find enriched markers per identified cluster and visualise these as a custom corrpplot.</i> |
|----------------|--|

Description

Find enriched markers per identified cluster and visualise these as a custom corrpplot.

Usage

```
plotSignatures(
  indata,
  assay = "scaled",
  clusterAssign = metadata(indata)[["Cluster"]],
  funcSummarise = function(x) mean(x, na.rm = TRUE),
  col = colorRampPalette(brewer.pal(9, "RdPu"))(100),
  labCex = 1,
  legendPosition = "right",
  legendCex = 1,
  labDegree = 90,
  verbose = TRUE
)
```

Arguments

| | |
|----------------|--|
| indata | A data-frame or matrix, or SingleCellExperiment object. If a data-frame or matrix, this should relate to expression data (cells as columns; genes as rows). If a SingleCellExperiment object, data will be extracted from an assay component named by assay. |
| assay | Name of the assay slot in indata from which data will be taken, assuming indata is a SingleCellExperiment object. |
| clusterAssign | A vector of cell-to-cluster assignments. This can be from any source but must align with your cells / variables. There is no check to ensure this when indata is not a SingleCellExperiment object. |
| funcSummarise | A mathematical function used to summarise expression per marker, per cluster. |
| col | colorRampPalette to be used for shading low-to-high expression. |
| labCex | cex (size) of the main plot labels. |
| legendPosition | position of legend. Can be one of 'top', 'right', 'bottom', 'left' |
| legendCex | cex (size) of the legend labels. |
| labDegree | Rotation angle of the main plot labels. |
| verbose | Boolean (TRUE / FALSE) to print messages to console or not. |

Details

Find enriched markers per identified cluster and visualise these as a custom corrplot. `plotSignatures` first collapses your input data's expression profiles from the level of cells to the level of clusters based on a mathematical function specified by `funcSummarise`. It then centers and scales the data range to be between -1 and +1 for visualisation purposes.

Value

A corrplot object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat <- jitter(matrix(
  MASS::rnegbin(rexp(1000, rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat) <- paste0('CD', 1:ncol(mat))
rownames(mat) <- paste0('cell', 1:nrow(mat))

u <- umap::umap(mat)$layout
colnames(u) <- c('UMAP1', 'UMAP2')
rownames(u) <- rownames(mat)
clus <- clusKNN(u)

plotSignatures(t(mat), clusterAssign = clus)
```

processFCS

Input, filter, normalise, and transform FCS expression data.

Description

Input, filter, normalise, and transform FCS expression data.

Usage

```
processFCS(
  files,
  assayname = "scaled",
  metadata = NULL,
  filter = TRUE,
  bgNoiseThreshold = 1,
  euclideanNormThreshold = 1,
  transformation = TRUE,
```

```

    transFun = function(x) asinh(x),
    asinhFactor = 5,
    downsample = 1e+05,
    downsampleVar = 0.1,
    colsDiscard = NULL,
    colsRetain = NULL,
    newColnames = NULL,
    emptyValue = TRUE,
    verbose = TRUE
)

```

Arguments

| | |
|------------------------|---|
| files | A vector of FCS files. |
| assayname | Name of the assay slot in which data will be stored. |
| metadata | Metadata associated with the FCS files specified in 'files'. A strict rule is enforced requiring that rownames(metadata) matches files in both name and order. |
| filter | Boolean (TRUE / FALSE) to enable filtering (per sample) for background signal / noise. |
| bgNoiseThreshold | Threshold for background noise. Used when filter == TRUE. |
| euclideanNormThreshold | Euclidean norm threshold for background noise. Used when filter == TRUE. |
| transformation | Boolean (TRUE / FALSE) to enable data transformation after filtering. |
| transFun | The function to apply (per sample) for transformation. Typically, for flow and mass cytometry, this is hyperbolic arc sine (asinh(x)). User can supply any function. |
| asinhFactor | The factor to apply when transforming via asinh(). For flow cytometry, this is usually 150; for mass cytometry and CyTOF, it is 5. Note that this is not used if the user has supplied their own function to transFun. |
| downsample | Downsample to this number of random variables. This is performed on the final merged dataset, i.e., after all samples have been bound together. NULL to disable. |
| downsampleVar | Downsample based on variance. Removes this proportion of cells based on lesser variance. This is applied per sample. If user wishes to apply this globally on the final merged dataset, then set this to 0 and remove based on variance manually. |
| colsDiscard | Columns to be removed from the final merged data. These names are literal and must match exactly. |
| colsRetain | Retain these columns only. This is the same as colsDiscard but in reverse. Technically, it is possible to activate both colsDiscard and colsRetain, but colsDiscard will be executed first. |
| newColnames | A named vector of new marker names to assign to each sample. The values of this vector should be the new marker names; the names of this vector should |

| | |
|------------|---|
| | represent the original marker names. This operation is performed AFTER any operation involving colsDiscard and colsRetain. |
| emptyValue | boolean (taken from ?flowCore::read.FCS indicating whether or not we allow an empty value for keyword values in TEXT segment. |
| verbose | Boolean (TRUE / FALSE) to print messages to console or not. |

Details

Input, filter, normalise, and transform FCS expression data.

Value

A SingleCellExperiment object.

Author(s)

Kevin Blighe <kevin@clinicalbioinformatics.co.uk>

Examples

```
# create random data that follows a negative binomial
mat1 <- jitter(matrix(
  MASS::rnegbin(rexp(50000), rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat1) <- paste0('CD', 1:ncol(mat1))
rownames(mat1) <- paste0('cell', 1:nrow(mat1))

mat2 <- jitter(matrix(
  MASS::rnegbin(rexp(50000), rate=.1), theta = 4.5),
  ncol = 20))
colnames(mat2) <- paste0('CD', 1:ncol(mat2))
rownames(mat2) <- paste0('cell', 1:nrow(mat2))

metadata <- data.frame(
  group = c('PB1', 'PB2'),
  row.names = c('mat1', 'mat2'),
  stringsAsFactors = FALSE)
```

Index

[basetheme](#), 2

[clusKNN](#), 4

[contourPlot](#), 6

[downsampleByVar](#), 9

[importData](#), 10

[markerEnrichment](#), 12

[markerExpression](#), 14

[markerExpressionPerCluster](#), 18

[metadataPlot](#), 20

[performUMAP](#), 24

[plotClusters](#), 25

[plotSignatures](#), 29

[processFCS](#), 30