Package 'CATALYST'

October 17, 2017

Type Package

Title Cytometry dATa anALYSis Tools

Version 1.1.1

Author Helena Lucia Crowell <crowellh@student.ethz.ch>,
Mark Robinson <mark.robinson@imls.uzh.ch>,
Vito Zanotelli <vito.zanotelli@uzh.ch>,
St<c3><a9>phane Chevrier, Bernd Bodenmiller

biocViews MassSpectrometry, Preprocessing, StatisticalMethod, SingleCell

Maintainer Helena Lucia Crowell <crowellh@student.ethz.ch>

Depends R (>= 3.4)

Description Mass cytometry (CyTOF) uses heavy metal isotopes rather than fluorescent tags as reporters to label antibodies, thereby substantially decreasing spectral overlap and allowing for examination of over 50 parameters at the single cell level. While spectral overlap is significantly less pronounced in CyTOF than flow cytometry, spillover due to detection sensitivity, isotopic impurities, and oxide formation can impede data interpretability. We designed CATALYST (Cytometry dATa anALYSis Tools) to provide a pipeline for preprocessing of cytometry data, including i) normalization using bead standards, ii) single-cell deconvolution, and iii) bead-based compensation.

Imports flowCore, ggplot2, graphics, grDevices, grid, gridExtra, matrixStats, methods, plotly, RColorBrewer, stats, utils

License GPL (>=2)

LazyData TRUE

VignetteBuilder knitr

RoxygenNote 6.0.1

Suggests BiocStyle, knitr, rmarkdown

NeedsCompilation no

R topics documented:

`applyCutoffs` *Single-cell debarcoding (2)*

Description

Applies separation and mahalanobies distance cutoffs.

Usage

```
applyCutoffs(x, ...)

## S4 method for signature 'dbFrame'
applyCutoffs(x, mhl_cutoff = 30, sep_cutoffs = NULL)
```

Arguments

`x`	a `dbFrame`.
`...`	optional arguments.
`mhl_cutoff`	mahalanobis distance threshold above which events should be unassigned. This argument will be ignored if the `mhl_cutoff` slot of the input `dbFrame` is specified.
`sep_cutoffs`	non-negative numeric of length one or same length as the number of rows in the `bc_key`. Specifies the distance separation cutoffs between positive and negative barcode populations above which events should be unassigned. If `NULL` (default), `applyCutoffs` will try to access the 'sep_cutoffs' slot of the supplied `dbFrame`.

Value

Will update the `bc_ids` and, if not already specified, `sep_cutoffs` and `mhl_cutoff` slots of the input `dbFrame`.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

References

Zunder, E.R. et al. (2015). Palladium-based mass tag cell barcoding with a doublet-filtering scheme and single-cell deconvolution algorithm. *Nature Protocols* **10**, 316-333.

Examples

```
data(sample_ff, sample_key)
re <- assignPrelim(x = sample_ff, y = sample_key)

# use global separation cutoff
applyCutoffs(x = re, sep_cutoffs = 0.4)

# estimate population-specific cutoffs
re <- estCutoffs(x = re)
applyCutoffs(x = re)
```

---

assignPrelim *Single-cell debarcoding (1)*

Description

Assigns a preliminary barcode ID to each event.

Usage

```
assignPrelim(x, y, ...)

## S4 method for signature 'flowFrame,data.frame'
assignPrelim(x, y, cofactor = 10,
  verbose = TRUE)

## S4 method for signature 'flowFrame,vector'
assignPrelim(x, y, cofactor = 10,
  verbose = TRUE)

## S4 method for signature 'character,data.frame'
assignPrelim(x, y, cofactor = 10,
  verbose = TRUE)

## S4 method for signature 'character,vector'
assignPrelim(x, y, cofactor = 10,
  verbose = TRUE)
```

Arguments

x	a flowFrame or character of an FCS file name.
y	the debarcoding scheme. A binary matrix with sample names as row names and numeric masses as column names OR a vector of numeric masses corresponding to barcode channels. When the latter is supplied, assignPrelim will create a scheme of the appropriate format internally.
...	optional arguments.
cofactor	cofactor used for asinh transformation.
verbose	logical. Should extra information on progress be reported? Defaults to TRUE.

Value

Returns a dbFrame containing measurement intensities, the debarcoding key, a numeric verctor of barcode IDs and separations between positive and negative barcode populations, and barcode intensities normalized by population.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

References

Zunder, E.R. et al. (2015). Palladium-based mass tag cell barcoding with a doublet-filtering scheme and single-cell deconvolution algorithm. *Nature Protocols* **10**, 316-333.

Examples

```
data(sample_ff, sample_key)
assignPrelim(x = sample_ff, y = sample_key)
```

---

compCytof *Compensate CyTOF experiment*

Description

Compensates a mass spectrometry based experiment using a provided spillover matrix, assuming a linear spillover in the experiment.

Usage

```
compCytof(x, y, ...)

## S4 method for signature 'flowFrame,matrix'
compCytof(x, y, out_path = NULL)

## S4 method for signature 'character,matrix'
compCytof(x, y, out_path = NULL)
```

Arguments

x	a flowFrame OR a character string specifying the location of FCS files that should be compensates.
y	a spillover matrix.
...	optional arguments.
out_path	a character string. If specified, compensated FCS files will be generated in this location. If x is a character string, file names will be inherited from uncompensated FCS files and given extension "_comped". Defaults to NULL.

Details

If the spillover matrix (SM) does not contain the same set of columns as the input experiment, it will be adapted according to the following rules:

1. columns present in the SM but not in the input data will be removed from it
2. non-metal columns present in the input but not in the SM will be added such that they do neither receive nor cause spill
3. metal columns that have the same mass as a channel present in the SM will receive (but not emit) spillover according to that channel
4. if an added channel could potentially receive spillover (as it has +/-1M or +16M of, or is of the same metal type as another channel measured), a warning will be issued as there could be spillover interactions that have been missed and may lead to faulty compensation

Value

Compensates the input flowFrame or, if x is a character string, all FCS files in the specified location. If out_path=NULL (the default), returns a flowFrame containing the compensated data. Otherwise, compensated data will be written to the specified location as FCS 3.0 standard files.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch> and Vito Zanotelli <vito.zanotelli@uzh.ch>

Examples

```
# get single-stained control samples
# get single-stained control samples
data(ss_exp)

# specify mass channels stained for
bc_ms <- c(139, 141:156, 158:176)

# debarcode
re <- assignPrelim(x = ss_exp, y = bc_ms)
re <- estCutoffs(x = re)
re <- applyCutoffs(x = re)
spillMat <- computeSpillmat(x = re)
compCytof(x = ss_exp, y = spillMat)
```

computeSpillmat *Compute spillover matrx*

Description

Computes the spillover matrix based on a priori identified single-positive popultions.

Usage

```
computeSpillmat(x, ...)

## S4 method for signature 'dbFrame'
computeSpillmat(x, method = "mean", trim = 0.08)
```

Arguments

x	a dbFrame.
...	optional arguments.
method	function to be used for computing spillover estimates. Defaults to mean.
trim	trim value used for estimation of spill values. Note that trim = 0.5 is equivalent to method = "median".

Value

Returns a square compensation matrix with dimensions and dimension names matching those of the input flowFrame. Spillover is assumed to be linear and is thence computed as the ratio of a positive barcode population's median or (trimmed) mean intensity in affected and spilling channel. Furthermore, on the basis of their additive nature, spillover values are computed independently for each interacting pair of channels. The current framework considers only potential (not all possible) interactions, that is, M+/-1 (detection sensitivity), same metals (isotopic impurites) and M+16M (oxide formation). By default, diagonal entries are set to 1.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

Examples

```
# get single-stained control samples
data(ss_exp)

# specify mass channels stained for
bc_ms <- c(139, 141:156, 158:176)

re <- assignPrelim(x = ss_exp, y = bc_ms)
re <- estCutoffs(x = re)
re <- applyCutoffs(x = re)
head(computeSpillmat(x = re))
```

concatFCS *FCS file concatination*

Description

Concatinates all input data.

Usage

```
concatFCS(x, ...)

## S4 method for signature 'flowSet'
concatFCS(x, out_path = NULL)

## S4 method for signature 'character'
concatFCS(x, out_path = NULL)

## S4 method for signature 'list'
concatFCS(x, out_path = NULL)
```

Arguments

x	can be either a flowSet, a list of flowFrames, a character specifying the location of the FCS files to be concatinated, or a vector of FCS file names.
...	optional arguments.
out_path	an optional character string. If specified, an FCS file of the concatinated data will be written to this location. If NULL (default), a flowFrame will be returned.

Value

a flowFrame containing measurement intensities of all input data or a character of the FCS file name.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

Examples

```
data(raw_data)
concatFCS(raw_data)
```

data *Example data sets*

Description

`raw_data` a `flowSet` with 3 experiments, each containing 2'500 raw measurements with a variation of signal over time. Samples were mixed with DVS beads capture by mass channels 140, 151, 153, 165 and 175.

`sample_ff` a `flowFrame` following a 6-choose-3 barcoding scheme where mass channels 102, 104, 105, 106, 108, and 110 were used for labeling such that each of the 20 individual barcodes are positive for exactly 3 out of the 6 barcode channels.

`sample_key` a `data.frame` of dimension 20 x 6 with sample names as row and barcode masses as column names. Contains a binary code of length 6 for each sample in `sample_ff`, e.g. 111000, as its unique identifier.

`ss_exp` a `flowFrame` with 20'000 events. Contains 36 single-antibody stained controls where beads were stained with antibodies captured by mass channels 139, 141 through 156, and 158 through 176, respectively, and pooled together.

`mp_cells` a `flowFrame` with 5000 spill-affected multiplexed cells and 39 measurement parameters.

Value

see descriptions above.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

Examples

```
### example data for normalization:
    # raw measurement data
    data(raw_data)

### example data for debarcoding:
    # 20 barcoded samples
    data(sample_ff)
    # 6-choose-3 barcoding scheme
    data(sample_key)

### example data for compensation:
    # single-stained control samples
    data(ss_exp)
    # multiplexed cells
    data(mp_cells)
```

dbFrame-class *Debarcoding frame class*

Description

This class represents the data returned by and used throughout debarcoding.

Details

Objects of class dbFrame hold all data required for debarcoding:

1. as the initial step of single-cell deconcolution, assignPrelim will return a dbFrame containing the input measurement data, barcoding scheme, and preliminary event assignments.
2. assignments will be made final by applyCutoffs. Optionally, population-specific separation cutoffs may be estimated by running estCutoffs prior to this.
3. plotYields, plotEvents and plotMahal aim to guide selection of devoncolution parameters and to give a sense of the resulting barcode assignment quality.

show(dbFrame) will display

- the dimensionality of the measurement data and number of barcodes
- current assignments in order of decreasing population size
- current separation cutoffs
- the average and per-population yield that will be achieven upon debarcoding

Slots

exprs a matrix containing raw intensities of the input flowFrame.

bc_key binary barcoding scheme with numeric masses as column names and samples names as row names OR a numeric vector of barcode masses.

bc_ids vector of barcode IDs. If a barcoding scheme is supplied, the respective binary code's row name, else, the mass of the respective barcode channel.

deltas numeric vector of separations between positive and negative barcode populations computed from normalized barcode intensities.

normed_bcs matrix containing normalized barcode intensities.

mhl_dists mahalanobis distances.

sep_cutoffs numeric vector of distance separation cutoffs between positive and negative barcode populations above which events will be unassigned.

mhl_cutoff non-negative and non-zero numeric value specifying the Mahalanobis distance below which events will be unassigned.

counts matrix of dimension (# barcodes)x(101) where each row contains the number of events within a barcode for which positive and negative populations are separated by a distance between in [0,0.01), ..., [0.99,1], respectively.

yields a matrix of dimension (# barcodes)x(101) where each row contains the percentage of events within a barcode that will be obtained after applying a separation cutoff of 0, 0.01, ..., 1, respectively.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

dbFrame-methods *Extraction and replacement methods for objects of class* dbFrame

Description

Methods for replacing and accessing slots in a dbFrame.

Usage

```
bc_key(x)

bc_ids(x)

deltas(x)

normed_bcs(x)

mhl_dists(x)

sep_cutoffs(x)

mhl_cutoff(x)

counts(x)

yields(x)

## S4 method for signature 'dbFrame'
exprs(object)

## S4 method for signature 'dbFrame'
bc_key(x)

## S4 method for signature 'dbFrame'
bc_ids(x)

## S4 method for signature 'dbFrame'
deltas(x)

## S4 method for signature 'dbFrame'
normed_bcs(x)

## S4 method for signature 'dbFrame'
mhl_dists(x)

## S4 method for signature 'dbFrame'
sep_cutoffs(x)

## S4 method for signature 'dbFrame'
mhl_cutoff(x)
```

```
## S4 method for signature 'dbFrame'
counts(x)

## S4 method for signature 'dbFrame'
yields(x)

## S4 replacement method for signature 'dbFrame,numeric'
mhl_cutoff(x) <- value

## S4 replacement method for signature 'dbFrame,ANY'
mhl_cutoff(x) <- value

## S4 replacement method for signature 'dbFrame,numeric'
sep_cutoffs(x) <- value

## S4 replacement method for signature 'dbFrame,ANY'
sep_cutoffs(x) <- value
```

Arguments

x, object	a dbFrame.
value	the replacement value.

Value

exprs extracts the raw data intensities.

bc_key extracts the barcoding scheme.

bc_ids extracts currently made event assignments.

deltas extracts barcode separations computed from normalized intensities. sep_cutoffs apply to these values (see applyCutoffs).

normed_bcs extracts normalized barcode intensities (see assignPrelim).

sep_cutoffs, sep_cutoffs<- extracts or replaces separation cutoffs. If option sep_cutoffs is not specified, these will be used by applyCutoffs. Replacement value must be a non- negative numeric with length one or same length as the number of barcodes.

mhl_cutoff, mhl_cutoff<- extracts or replaces the Mahalanobis distance threshold above which events are to be unassigned. Replacement value must be a single non-negative and non-zero numeric.

counts extract the counts matrix (see dbFrame).

yields extract the yields matrix (see dbFrame).

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

Examples

```
data(sample_ff, sample_key)
re <- assignPrelim(x = sample_ff, y = sample_key)

# set global cutoff parameter
sep_cutoffs(re) <- 0.4
```

```
re <- applyCutoffs(x = re)

# subset a specific population, e.g. A1: 111000
a1 <- bc_ids(re) == "A1"
head(exprs(sample_ff[a1, ]))

# subset unassigned events
unassigned <- bc_ids(re) == 0
head(exprs(sample_ff[unassigned, ]))
```

`estCutoffs` *Estimation of distance separation cutoffs*

Description

For each barcode, estimates a cutoff parameter for the distance between positive and negative barcode populations.

Usage

```
estCutoffs(x, ...)

## S4 method for signature 'dbFrame'
estCutoffs(x, verbose = TRUE)
```

Arguments

`x`	a `dbFrame`.
`...`	optional arguments.
`verbose`	logical. Should extra information on progress be reported? Defaults to TRUE.

Value

Will update the `sep_cutoffs`, `mhl_cutoff`, `counts` and `yields` slots of the input `dbFrame` and return the latter.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

Examples

```
data(sample_ff, sample_key)
re <- assignPrelim(x = sample_ff, y = sample_key)
estCutoffs(x = re)
```

estTrim *Estimation of optimal trim value*

Description

Estimates a trim value that will minimize the sum over squared popultion- and channel-wise squared medians upon compensation.

Usage

```
estTrim(x, ...)

## S4 method for signature 'dbFrame'
estTrim(x, min = 0.05, max = 0.2, step = 0.01,
  out_path = NULL, name_ext = NULL)
```

Arguments

x	a dbFrame.
...	optional arguments.
min, max, step	specifies sequence of trim values for which compensation should be evaluated.
out_path	specifies in which location output plot is to be generated. Defaults to NULL.
name_ext	a character string. If specified, will be appended to the output plot's name. Defaults to NULL.

Value

For each value along seq(min, max, step), estTrim will call computeSpillmat with method = "mean" and the respective trim parameter. Returned will be an interactive plot displaying channel-wise medians upon compensation, and the mean squared deviation from 0. Each point is labeled with the respective interacting channels.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

Examples

```
# get single-stained control samples
data(ss_exp)

# specify mass channels stained for
bc_ms <- c(139, 141:156, 158:176)

re <- assignPrelim(x = ss_exp, y = bc_ms)
re <- estCutoffs(x = re)
re <- applyCutoffs(x = re)
estTrim(x = re, min = 0.02, max = 0.14, step = 0.02)
```

normCytof *Bead-based normalization*

Description

an implementation of Finck et al.'s normalization of mass cytometry data using bead standards with automated bead gating.

Usage

```
normCytof(x, y, ...)

## S4 method for signature 'flowFrame'
normCytof(x, y, out_path = NULL, remove_beads = TRUE,
  norm_to = NULL, k = 500, trim = 5, verbose = TRUE)

## S4 method for signature 'character'
normCytof(x, y, out_path = NULL, remove_beads = TRUE,
  norm_to = NULL, k = 500, trim = 5, verbose = TRUE)
```

Arguments

x	a flowFrame or character of the FCS file to be normalized.
y	"dvs" (for bead masses 140, 151, 153, 165, 175) or "beta" (for bead masses 139, 141, 159, 169, 175) or a numeric vector of bead masses.
...	optional arguments.
out_path	a character string. If specified, outputs will be generated in this location. If NULL (the default), normCytof will return a flowFrame of the normalized data (if remove=FALSE) or a flowSet containing normalized cells and beads (if remove=TRUE).
remove_beads	logical. If TRUE (the default) beads will be removed and normalized cells and beads returned separately.
norm_to	a flowFrame or character of an FCS file from which baseline values should be computed and to which the input data should be normalized.
k	integer width of the median window used for bead smoothing.
trim	a single non-negative numeric. A *median* +/- ... *mad* rule is applied to the preliminary population of bead events to remove bead-bead doublets and low signal beads prior to estimating normalization factors.
verbose	logical. Should extra information on progress be reported? Defaults to TRUE.

Value

if out_path=NULL (the default) a flowFrame of the normalized data (if remove=FALSE) or flowSet containing normalized cells and beads (if remove=TRUE). Else, a character of the location where output FCS files and plots have been generated.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

References

Finck, R. et al. (2013). Normalization of mass cytometry data with bead standards. *Cytometry A* **83A**, 483-494.

Examples

```
data(raw_data)
ff <- concatFCS(raw_data)
normCytof(x = ff, y = "dvs", k=300)
```

`outFCS` *Write population-wise FCS files*

Description

Writes an FCS file for each sample from a dbFrame.

Usage

```
outFCS(x, out_path = tempdir(), ...)

## S4 method for signature 'dbFrame'
outFCS(x, out_path = tempdir(), out_nms = NULL,
  verbose = TRUE)
```

Arguments

`x`	a `dbFrame`.
`out_path`	character string. Specifies in which location output files are to be generated.
`...`	optional arguments.
`out_nms`	an optional character string. Either the name of a 2 column CSV table with sample IDs and desired output file names, or a vector of length `nrow(bc_key(x))` ordered as the samples in the barcoding scheme. If NULL (default), sample IDs will be used as file names.
`verbose`	if TRUE (default), a warning is given about populations for which no FCS files have been generated.

Details

Creates a separate FCS file for each barcode population. If `out_nms` is NULL (the default), files will be named after the barcode population's ID in the `bc_key` slot of the input `dbFrame`; unassigned events will be written to "unassigned.fcs", and no output is generated for populations with less than 10 event assignments.

Value

a character of the output path.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

Examples

```
data(sample_ff, sample_key)
re <- assignPrelim(x = sample_ff, y = sample_key)
re <- estCutoffs(x = re)
re <- applyCutoffs(x = re)
outFCS(x = re, out_path = file.path(tempdir()))
```

---

plotEvents *Event plot*

Description

Shows normalized barcode intensities for a given barcode.

Usage

```
plotEvents(x, ...)

## S4 method for signature 'dbFrame'
plotEvents(x, which = "all", n_events = 100,
  out_path = NULL, name_ext = NULL)
```

Arguments

x	a dbFrame.
...	optional arguments.
which	"all", numeric or character. Specifies which barcode(s) to plot. Valid values are IDs that occur as row names in the bc_key of the supplied dbFrame, or 0 for unassigned events. Defaults to "all".
n_events	numeric. Specifies number of events to plot. Defaults to 100.
out_path	a character string. If specified, outputs will be generated in this location. Defaults to NULL.
name_ext	a character string. If specified, will be appended to the plot's name. Defaults to NULL.

Value

plots intensities normalized by population for each barcode specified by which: Each event corresponds to the intensities plotted on a vertical line at a given point along the x-axis. Events are scaled to the 95% quantile of the population it has been assigned to. Barcodes with less than 50 event assignments will be skipped; it is strongly recoomended to remove such populations or reconsider their separation cutoffs.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

References

Zunder, E.R. et al. (2015). Palladium-based mass tag cell barcoding with a doublet-filtering scheme and single-cell deconvolution algorithm. *Nature Protocols* **10**, 316-333.

Examples

```
data(sample_ff, sample_key)

# view preliminary assignments
re <- assignPrelim(x = sample_ff, y = sample_key)
plotEvents(x = re, which = "D1", n_events = 1000)

# apply deconvolution parameters
re <- estCutoffs(re)
re <- applyCutoffs(x = re)
plotEvents(x = re, which = "D1", n_events = 500)
```

---

plotMahal *Biaxial plot*

Description

Histogram of counts and plot of yields as a function of separation cutoffs.

Usage

```
plotMahal(x, ...)

## S4 method for signature 'dbFrame'
plotMahal(x, which, cofactor = 50, out_path = NULL,
  name_ext = NULL)
```

Arguments

x	a dbFrame.
...	optional arguments.
which	specifies which barcode to plot.
cofactor	cofactor used for asinh transformation.
out_path	a character string. If specified, outputs will be generated in this location. Defaults to NULL.
name_ext	a character string. If specified, will be appended to the plot's name. Defaults to NULL.

Value

plots all inter-barcode interactions for the population specified by argument which. Events are colored by their Mahalanobis distance.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

References

Zunder, E.R. et al. (2015). Palladium-based mass tag cell barcoding with a doublet-filtering scheme and single-cell deconvolution algorithm. *Nature Protocols* **10**, 316-333.

Examples

```
data(sample_ff, sample_key)
re <- assignPrelim(x = sample_ff, y = sample_key)
re <- estCutoffs(x = re)
re <- applyCutoffs(x = re)
plotMahal(x = re, which = "B3")
```

---

plotSpillmat *Spillover matrix heat map*

Description

Generates a heat map of the spillover matrix annotated with estimated spill percentages.

Usage

```
plotSpillmat(bc_ms, SM, annotate = TRUE, palette = NULL)
```

Arguments

bc_ms	a vector of numeric masses corresponding to barcode channels.
SM	spillover matrix returned from computeSpillmat.
annotate	logical. If TRUE (default), spill percentages are shown inside bins and rows/columns are annotated with the total amount of spill caused/received.
palette	an optional vector of colors to interpolate.

Value

plots estimated spill percentages as a heat map. Colours are ramped to the highest spillover value present

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

Examples

```
# get single-stained control samples
data(ss_exp)

# specify mass channels stained for
bc_ms <- c(139, 141:156, 158:176)

re <- assignPrelim(x = ss_exp, y = bc_ms)
re <- estCutoffs(x = re)
re <- applyCutoffs(x = re)
spillMat <- computeSpillmat(x = re)
plotSpillmat(bc_ms = bc_ms, SM = spillMat)
```

`plotYields` *Yield plot*

Description

Distribution of barcode separations and yields as a function of separation cutoffs.

Usage

```
plotYields(x, ...)

## S4 method for signature 'dbFrame'
plotYields(x, which = 0, annotate = TRUE,
  legend = TRUE, out_path = NULL, name_ext = NULL)
```

Arguments

`x`	a `dbFrame`.
`...`	optional arguments.
`which`	0, numeric or character. Specifies which barcode(s) to plot. Valid values are IDs that occur as row names in the `bc_key` of the supplied `dbFrame`; 0 (the default) will generate a summary plot with all barcodes.
`annotate`	logical. If TRUE (default) and the `sep_cutoffs` slot of the supplied `dbFrame` is not empty, vertical lines will be drawn at cutoff values and the resulting yield will be included in the plot title.
`legend`	logical. Specifies if a legend should be included. This will only affect the summary plot (`which=0`).
`out_path`	a character string. If specified, outputs will be generated in this location. Defaults to NULL.
`name_ext`	a character string. If specified, will be appended to the plot's name. Defaults to NULL.

Details

The overall yield that will be achieved upon application of the specified set of separation cutoffs is indicated in the summary plot. Respective separation thresholds and their resulting yields are included in each barcode's plot. The separation cutoff value should be chosen such that it appropriately balances confidence in barcode assignment and cell yield.

Value

plots the distribution of barcode separations and yields upon debarcoding as a function of separation cutoffs. If available, currently used separation cutoffs as well as their resulting yields will be indicated in the plot‘s main title.

Author(s)

Helena Lucia Crowell <crowellh@student.ethz.ch>

References

Zunder, E.R. et al. (2015). Palladium-based mass tag cell barcoding with a doublet-filtering scheme and single-cell deconvolution algorithm. *Nature Protocols* **10**, 316-333.

Examples

```
data(sample_ff, sample_key)
re <- assignPrelim(x = sample_ff, y = sample_key)
re <- estCutoffs(x = re)

# all barcodes summary plot
plotYields(x = re, which = 0)

# plot for specific sample
plotYields(x = re, which = "C1")
```

Index