

Package ‘CATALYST’

April 11, 2018

Type Package

Title Cytometry dATa anALYSis Tools

Version 1.2.0

Author Helena Lucia Crowell <crowellh@student.ethz.ch>, Mark D Robinson <mark.robinson@imls.uzh.ch>, Vito RT Zantelli <vito.zantelli@uzh.ch>, Stéphane Chevrier, Bernd Bodenmiller

biocViews MassSpectrometry, Preprocessing, StatisticalMethod, SingleCell, Normalization

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 drc, flowCore, ggplot2, graphics, grDevices, grid, gridExtra, matrixStats, methods, plotly, RColorBrewer, reshape2, stats, utils

Suggests BiocStyle, knitr, rmarkdown, testthat

VignetteBuilder knitr

RoxygenNote 6.0.1

License GPL (>=2)

LazyData TRUE

Encoding UTF-8

NeedsCompilation no

R topics documented:

applyCutoffs	2
assignPrelim	3
compCytof	4
computeSpillmat	6

concatFCS	7
data	8
dbFrame-class	9
dbFrame-methods	10
estCutoffs	12
estTrim	13
normCytof	14
outFCS	16
outFrames	17
plotEvents	18
plotMahal	19
plotSpillmat	20
plotYields	21

Index	23
--------------	-----------

applyCutoffs	<i>Single-cell debarcoding (2)</i>
--------------	------------------------------------

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	<i>Single-cell debarcoding (1)</i>
--------------	------------------------------------

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 vector 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)
```

compCytotf

Compensate CyTOF experiment

Description

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

Usage

```
compCytotf(x, y, ...)

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

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

## S4 method for signature 'ANY,data.frame'
compCytotf(x, y, out_path = NULL)
```

Arguments

x	a <code>flowFrame</code> OR a character string specifying the location of FCS files that should be compensated.
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 Zanutelli <vito.zanutelli@uzh.ch>

Examples

```
# 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	<i>Compute spillover matrix</i>
-----------------	---------------------------------

Description

Computes a spillover matrix from a priori identified single-positive populations.

Usage

```
computeSpillmat(x, ...)

## S4 method for signature 'dbFrame'
computeSpillmat(x, method = "default",
  interactions = "default", trim = 0.5, th = 1e-05)
```

Arguments

x	a dbFrame .
...	optional arguments.
method	function to be used for computing spillover estimates (see below for details).
interactions	"default" or "all". Specifies which interactions spillover should be estimated for. The default exclusively takes into consideration interactions that are sensible from a chemical and physical point of view (see below for more details).
trim	trim value used for estimation of spill values. Note that trim = 0.5 is equivalent to using medians.
th	a single non-negative numeric. Specifies a threshold value below which spill estimates will be set to 0.

Details

The default method estimates the spillover as the median ratio between the unstained spillover receiving and the stained spillover emitting channel in the corresponding single stained populations.

method = "classic" will compute the slope of a line through the medians (or trimmed means) of stained and unstained populations. The medians (or trimmed means) computed from events that are i) negative in the respective channels; and, ii) not assigned to interacting channels; and, iii) not unassigned are subtracted as to account for background.

interactions="default" considers only expected interactions, that is, M+/-1 (detection sensitivity), M+16 (oxide formation) and channels measuring metals that are potentially contaminated by isotopic impurities (see reference below and [isotope_list](#)).

interaction="all" will estimate spill for all $n \times n - n$ interactions, where n denotes the number of single-color controls (= nrow(bc_key(re))).

Value

Returns a square compensation matrix with dimensions and dimension names matching those of the input flowFrame. Spillover is assumed to be linear, and, on the basis of their additive nature, spillover values are computed independently for each interacting pair of channels.

Author(s)

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

References

Coursey, J.S., Schwab, D.J., Tsai, J.J., Dragoset, R.A. (2015). Atomic weights and isotopic compositions, (available at <http://physics.nist.gov/Comp>).

Examples

```
# get single-stained control samples
data(ss_exp)
# specify mass channels stained for
bc_ms <- c(139, 141:156, 158:176)
# debarcode single-positive populations
re <- assignPrelim(x = ss_exp, y = bc_ms)
re <- estCutoffs(x = re)
re <- applyCutoffs(x = re)
head(computeSpillmat(x = re))
```

concatFCS

FCS file concatenation

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 concatenated, or a vector of FCS file names.
...	optional arguments.
out_path	an optional character string. If specified, an FCS file of the concatenated 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.

`isotope_list` a named list of isotopic compositions for all elements within 75 through 209 u corresponding to the CyTOF mass range at the time of writing (see reference).

Value

see descriptions above.

Author(s)

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

References

Coursey, J.S., Schwab, D.J., Tsai, J.J., Dragoset, R.A. (2015). Atomic weights and isotopic compositions, (available at <http://physics.nist.gov/Comp>).

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 deconvolution, `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 deconvolution 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 achieved 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)
```

Arguments

x a `dbFrame`.

... optional arguments.

Details

For the estimation of sample-specific cutoff parameters, we fit a three-parameter log-logistic function to the yields function. As an adequate cutoff estimate, we target a point which approximately marks the end of the plateau regime and on-set of yield decline. By default, we compute this as the first minimum of the fifth derivative (`deriv=5`). However, depending on the overall doublet-singlet separation, and how bimodal the count distribution is, another derivate may provide a better estimate. As a general guideline, higher values of `deriv` will shift the computed minimum towards the left and yield more liberal (low) cutoffs, while a low `deriv` will shift it towards the inflection point of the yields functions, arriving at more stringent (high) estimates.

Value

Will update the `sep_cutoffs` slot of the input `dbFrame` and return the latter.

Author(s)

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

Examples

```
data(sample_ff, sample_key)
# assign preliminary IDs
re <- assignPrelim(x = sample_ff, y = sample_key)
# estimate separation cutoffs
re <- estCutoffs(x = re)
# view exemplary estimate
plotYields(re, "A1")
```

 estTrim

Estimation of optimal trim value

Description

Estimates a trim value that will minimize the sum over squared population- 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,
        method = "default", interactions = "default", 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.

method	function to be used for computing spillover estimates. Valid options are "default" or "classic" (see computeSpillmat for details)
interactions	"default" or "all". Specifies which interactions spillover should be estimated for. The default exclusively takes into consideration interactions that are sensible from a chemical and physical point of view (see computeSpillmat for more detail).
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, max = 0.12, step = 0.02, method = "classic")
```

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 <i>median +/- ... mad</i> 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	<i>Write population-wise FCS files</i>
--------	--

Description

Writes an FCS file for each sample from a dbFrame.

Usage

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

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

Arguments

x	a dbFrame .
y	a flowFrame containing the original measurement and meta data.
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, y = sample_ff)
```

outFrames	<i>Population-wise flowFrames from a dbFrame</i>
-----------	--

Description

Returns a flowSet or list of flowFrames from a [dbFrame](#). Each flowFrame will contain the subset of events that have been assigned to the same ID.

Usage

```
outFrames(x, ...)  
  
## S4 method for signature 'dbFrame'  
outFrames(x, return = "flowSet", which = "assigned")
```

Arguments

x	a dbFrame .
...	optional arguments.
return	"flowSet" or "list". Specifies the output type.
which	Specifies which barcode(s) to include. "assigned" (if the population of unassigned events should be excluded), "all" (if the latter should be included), or a numeric or character specifying a subset of populations. Valid values are IDs that occur as row names in the bc_key of the supplied dbFrame . Defaults to "assigned".

Details

Creates a separate [flowFrame](#) for each barcode population and, if desired, the population of unassigned events.

Value

a [flowSet](#) or list of [flowFrames](#).

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)  
outFrames(x = re, return = "list", which = c("B1", "D4"))
```

plotEvents	<i>Event plot</i>
------------	-------------------

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 <code>bc_key</code> 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 recommended 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	<i>Biaxial plot</i>
-----------	---------------------

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	<i>Spillover matrix heat map</i>
--------------	----------------------------------

Description

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

Usage

```
plotSpillmat(bc_ms, SM, annotate = TRUE, palette = NULL, out_path = NULL,
             name_ext = 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.
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 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 <code>bc_key</code> of the supplied dbFrame ; 0 (the default) will generate a summary plot with all barcodes.
annotate	logical. If TRUE (default) and the <code>sep_cutoffs</code> 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 (<code>which=0</code>).
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

- applyCutoffs, [2, 9, 11, 12](#)
- applyCutoffs, dbFrame-method
 - (applyCutoffs), [2](#)
- assignPrelim, [3, 9, 11](#)
- assignPrelim, character, data.frame-method
 - (assignPrelim), [3](#)
- assignPrelim, character, vector-method
 - (assignPrelim), [3](#)
- assignPrelim, flowFrame, data.frame-method
 - (assignPrelim), [3](#)
- assignPrelim, flowFrame, vector-method
 - (assignPrelim), [3](#)

- bc_ids (dbFrame-methods), [10](#)
- bc_ids, dbFrame-method
 - (dbFrame-methods), [10](#)
- bc_key (dbFrame-methods), [10](#)
- bc_key, dbFrame-method
 - (dbFrame-methods), [10](#)

- compCytof, [4](#)
- compCytof, ANY, data.frame-method
 - (compCytof), [4](#)
- compCytof, character, matrix-method
 - (compCytof), [4](#)
- compCytof, flowFrame, matrix-method
 - (compCytof), [4](#)
- computeSpillmat, [6, 14](#)
- computeSpillmat, dbFrame-method
 - (computeSpillmat), [6](#)
- concatFCS, [7](#)
- concatFCS, character-method (concatFCS),
[7](#)
- concatFCS, flowSet-method (concatFCS), [7](#)
- concatFCS, list-method (concatFCS), [7](#)
- counts (dbFrame-methods), [10](#)
- counts, dbFrame-method
 - (dbFrame-methods), [10](#)

- data, [8](#)
- dbFrame, [2, 4, 6, 10–13, 16–19, 21](#)
- dbFrame (dbFrame-class), [9](#)
- dbFrame-class, [9](#)
- dbFrame-methods, [10](#)

- deltas (dbFrame-methods), [10](#)
- deltas, dbFrame-method
 - (dbFrame-methods), [10](#)

- estCutoffs, [9, 12](#)
- estCutoffs, dbFrame-method (estCutoffs),
[12](#)
- estTrim, [13](#)
- estTrim, dbFrame-method (estTrim), [13](#)
- exprs (dbFrame-methods), [10](#)
- exprs, dbFrame-method (dbFrame-methods),
[10](#)

- flowFrame, [3, 5, 7, 8, 15–17](#)
- flowSet, [7, 8, 15, 17](#)

- isotope_list, [6](#)
- isotope_list (data), [8](#)

- mhl_cutoff (dbFrame-methods), [10](#)
- mhl_cutoff, dbFrame-method
 - (dbFrame-methods), [10](#)
- mhl_cutoff<- (dbFrame-methods), [10](#)
- mhl_cutoff<-, dbFrame, ANY-method
 - (dbFrame-methods), [10](#)
- mhl_cutoff<-, dbFrame, numeric-method
 - (dbFrame-methods), [10](#)
- mhl_dists (dbFrame-methods), [10](#)
- mhl_dists, dbFrame-method
 - (dbFrame-methods), [10](#)
- mp_cells (data), [8](#)

- normCytof, [14](#)
- normCytof, character-method (normCytof),
[14](#)
- normCytof, flowFrame-method (normCytof),
[14](#)
- normed_bcs (dbFrame-methods), [10](#)
- normed_bcs, dbFrame-method
 - (dbFrame-methods), [10](#)

- outFCS, [16](#)
- outFCS, dbFrame, flowFrame-method
 - (outFCS), [16](#)
- outFrames, [17](#)

outFrames, dbFrame-method (outFrames), 17

plotEvents, 9, 18
plotEvents, dbFrame-method (plotEvents),
18

plotMahal, 9, 19
plotMahal, dbFrame-method (plotMahal), 19

plotSpillmat, 20

plotYields, 9, 21
plotYields, dbFrame-method (plotYields),
21

raw_data (data), 8

sample_ff (data), 8

sample_key (data), 8

sep_cutoffs (dbFrame-methods), 10
sep_cutoffs, dbFrame-method
(dbFrame-methods), 10

sep_cutoffs<- (dbFrame-methods), 10

sep_cutoffs<- , dbFrame, ANY-method
(dbFrame-methods), 10

sep_cutoffs<- , dbFrame, numeric-method
(dbFrame-methods), 10

ss_exp (data), 8

yields (dbFrame-methods), 10
yields, dbFrame-method
(dbFrame-methods), 10