

Package ‘flowPloidy’

April 14, 2017

Title Analyze flow cytometer data to determine sample ploidy

Version 1.0.0

Author Tyler Smith <tyler@plantarum.ca>

Maintainer Tyler Smith <tyler@plantarum.ca>

Description Determine sample ploidy via flow cytometry histogram analysis.
Reads Flow Cytometry Standard (FCS) files via the flowCore bioconductor package, and provides functions for determining the DNA ploidy of samples based on internal standards.

biocViews FlowCytometry, GUI, Regression, Visualization

URL <https://github.com/plantarum/flowPloidy>

BugReports <https://github.com/plantarum/flowPloidy/issues>

Depends R (>= 3.3)

License GPL-3

LazyData true

Imports flowCore, car, caTools, knitr, rmarkdown, minpack.lm, shiny, methods, graphics, stats, utils

Suggests flowPloidyData

RoxygenNote 5.0.1

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

browseFlowHist	2
fhAnalyze	2
findPeaks	3
FlowHist	5
gauss	7
pickInit	7
plot.FlowHist	8
plotFH	9
setBins	9
tabulateFlowHist	10
updateFlowHist	11
viewFlowChannels	12

Index**13**

browseFlowHist	<i>browseFlowHist</i>
----------------	-----------------------

Description

Visually assess histogram fits

Usage

```
browseFlowHist(flowList, debug = FALSE)
```

Arguments

flowList	a list of FlowHist objects
debug	boolean, turns on debugging messages

Details

Visually assess histogram fits

Value

Returns the list of FlowHist objects, updated by any changes made in the GUI.

Author(s)

Tyler Smith

Examples

```
library(flowPloidyData)
batch1 <- batchFlowHist(flowPloidyFiles, channel = "FL3.INT.LIN")
## Not run:
batch1 <- browseFlowHist(batch1)

## End(Not run)
```

fhAnalyze	<i>fhAnalyze</i>
-----------	------------------

Description

Complete non-linear regression analysis of FlowHist histogram data

Usage

```
fhAnalyze(fh)
```

Arguments

fh a FlowHist object

Details

Completes the NLS analysis, and calculates the modelled events and CVs for the result.

Value

a FlowHist object with the analysis (nls, counts, cv, RCS) slots filled.

Author(s)

Tyler Smith

See Also

[FlowHist](#)

Examples

```
library(flowPloidyData)
fh1 <- FlowHist(file = flowPloidyFiles[1], channel = "FL3.INT.LIN")
fh1 <- fhAnalyze(fh1)
```

findPeaks

findPeaks

Description

Locate potential peaks in histogram data

Usage

```
findPeaks(fh, window, smooth = window/2)
```

```
cleanPeaks(fh, window)
```

Arguments

fh a FlowHist object

window an integer, the width of the moving window to use in identifying local maxima via `caTools::runmax`

smooth an integer, the width of the moving window to use in removing noise via `caTools::runmean`

Details

Peaks are defined as local maxima in the vector of values, using a moving window. Note that these are used in the context of finding starting values - accuracy isn't important, we just need something 'close-enough' that the nls algorithm will be able to find the correct value.

Utility functions for use internally by flowPloidy; not exported and won't be visible to users. Usually invoked from within FlowHist.

Note that there is a trade-off between accuracy in detected peaks, and avoiding noise. Increasing the value of `smooth` will reduce the amount of 'noise' that is included in the peak list. However, increasing smoothing shifts the location of the actual peaks. Most of the time the default values provide an acceptable compromise, given we only need to get 'close enough' for the NLS optimization to find the true parameter values. If you'd like to explore this, the internal (unexported) function `fhPeakPlot` may be useful.

`cleanPeaks` filters the output of `findPeaks` to:

- remove duplicates, ie., peaks with the same intensity that occur within window positions of each other. Otherwise, `findPeaks` will consider noisy peaks without a single highest point to be multiple distinct peaks.
- drop G2 peaks. In some cases the G2 peak for one sample will have greater intensity than the G1 peak for another sample. We correct for this by removing detected peaks with means close to twice that of other peaks.
- ignore noise, by removing peaks with intensity < 40. A somewhat arbitrary value. It's tricky to deal with this issue when the debris field is large.

Value

Returns a matrix with two columns:

mean the index position of each potential peak

height the height (intensity) of the peak at that index position

Author(s)

Tyler Smith

Examples

```
## Not run:
set.seed(123)
test.dat <- (cumsum(runif(1000, min = -1)))
plot(test.dat, type = 'l')
test.peaks <- flowPloidy::findPeaks(test.dat, window = 20)
points(test.peaks, col = 'red', cex = 2)

## End(Not run)
```

FlowHist

*FlowHist***Description**

Creates a FlowHist object from an FCS file, setting up the histogram data for analysis.

Usage

```
FlowHist(file, channel, bins = 256, window = 20, smooth = 20,
  pick = FALSE, linearity = "variable", debris = "SC", opts = list(),
  analyze = TRUE)
```

```
batchFlowHist(files, channel, bins = 256, verbose = TRUE, window = 20,
  smooth = 20, linearity = "variable", debris = "SC")
```

Arguments

file	character, the name of the file to load
channel	character, the name of the data column to use
bins	integer, the number of bins to use to aggregate events into a histogram
window	the width of the moving window used to identify local maxima for peak detection via <code>caTools::runmax</code>
smooth	the width of the moving window used to reduce noise in the histogram via <code>caTools::runmean</code>
pick	boolean; if TRUE, the user will be prompted to select peaks to use for starting values. Otherwise (the default), starting values will be detected automatically.
linearity	character, either "variable", the default, or "fixed". If "fixed", linearity is fixed at 2; if "variable", linearity is fit as a model parameter.
debris	character, either "SC", the default, or "MC", to set the debris model component to the Single-Cut or Multi-Cut models.
opts	list, currently not used, but maybe in future as a way to test additional model options
analyze	boolean, if TRUE the model will be analyzed immediately
files	character, a vector of file names to load
verbose	boolean; if TRUE, <code>histBatch</code> will list files as it processes them.

Details

Starting with a `flowFrame` object, read from a FCS file, `FlowHist` will:

1. Extract the intensity data from channel.
2. Remove the top bin, which contains off-scale readings we ignore in the analysis.
3. aggregates the raw data into the desired number of bins, as specified with the `bins` argument. The default is 256, but you may also try 128 or 512. Any integer is technically acceptable, but I wouldn't stray from the default without a good reason.

4. identify model components to include. All FlowHist objects will have the single-cut debris model and the G1 peak for sample A, and the broadened rectangle for the S-phase of sample A. Depending on the data, additional components for the G2 peak and sample B (G1, G2, s-phase) may also be added.
5. Build the NLS model. All the components are combined into a single model.
6. Identify starting values for Gaussian (G1 and G2 peaks) model components. For reasonably clean data, the built-in peak detection is fine. You can evaluate this by plotting the FlowHist object with the argument `init = TRUE`. If it doesn't look good, you can play with the `window` and `smooth` arguments (which is tedious!), or pick the peaks visually yourself with `pick = TRUE`.

Value

FlowHist returns a FlowHist object.

batchFlowHist returns a list of FlowHist objects.

Slots

`raw` a flowFrame object containing the raw data from the FCS file

`channel` character, the name of the data column to use

`bins` integer, the number of bins to use to aggregate events into a histogram

`histdata` data.frame, the columns are the histogram bin number (`xx`), florescence intensity (`intensity`), and the raw single-cut debris model values (`SCVals`, used in model fitting). Additional columns may be added if/when I add gating, so refer to columns by name, not position.

`peaks` matrix, containing the coordinates used for peaks when calculcating initial parameter values.

`comps` a list of modelComponent objects included for these data.

`model` the function (built from `comps`) to fit to these data.

`init` a list of initial parameter estimates to use in fitting the model.

`nls` the nls object produced by the model fitting

`counts` a list of cells counted in each peak of the fitted model

`CV` a list of the coefficients of variation for each peak in the fitted model.

`RCS` numeric, the residual chi-square for the fitted model.

Author(s)

Tyler Smith

Examples

```
library(flowPloidyData)
fh1 <- FlowHist(file = flowPloidyFiles[1], channel = "FL3.INT.LIN")
fh1
library(flowPloidyData)
batch1 <- batchFlowHist(flowPloidyFiles, channel = "FL3.INT.LIN")
batch1
```

gauss	<i>Gaussian model components</i>
-------	----------------------------------

Description

Components for modeling Gaussian features in flow histograms

Arguments

a1, a2, b1, b2	area parameters
Ma, Mb	curve mean parameter
Sa, Sb	curve standard deviation parameter
xx	vector of histogram intensities

Details

Typically the complete models will contain fA1 and fB2, which model the G1 peaks of the sample and the standard. In most cases, they will also contain fA2 and fB2, which model the G2 peaks. The G2 peaks are linked to the G1 peaks, in that they require some of the parameters from the G1 peaks as well (mean and standard deviation).

Value

NA

Author(s)

Tyler Smith

pickInit	<i>Interactively select model starting values</i>
----------	---

Description

Prompts the user to select the peaks to use as initial values for non-linear regression on a plot of the histogram data.

Usage

```
pickInit(fh)
```

Arguments

fh	A flowHist object
----	-------------------

Details

The raw histogram data are plotted, and the user is prompted to select the peak positions to use as starting values in the NLS procedure. This is useful when the automated peak-finding algorithm fails to discriminate between overlapping peaks, or is confused by noise.

The normal use, pickPeaks is called from pickInit, rather than directly by the user.

Value

pickInit returns the flowHist object with its initial value slot updated.

pickPeaks returns a matrix with each peak as a row, with the mean (position) in the first column, and the height (intensity) in the second column.

Author(s)

Tyler Smith

Examples

```
library(flowPloidyData)
fh2 <- FlowHist(file = flowPloidyFiles[12], channel = "FL3.INT.LIN")
plot(fh2, init = TRUE) ## automatic peak estimates
## Not run:
fh2 <- pickInit(fh2)  ## hand-pick peak estimates

## End(Not run)
plot(fh2, init = TRUE) ## revised starting values
```

plot.FlowHist

Plot histograms for FlowHist objects

Description

Plot histograms for FlowHist objects

Usage

```
## S3 method for class 'FlowHist'
plot(x, init = FALSE, nls = TRUE, comps = TRUE, ...)
```

Arguments

x	a FlowHist object
init	boolean; if TRUE, plot the regression model using the initial parameter estimates over the raw data.
nls	boolean; if TRUE, plot the fitted regression model over the raw data (i.e., using the final parameter values)
comps	boolean; if TRUE, plot the individual model components over the raw data.
...	additional arguments passed on to plot()

Value

Not applicable

Author(s)

Tyler Smith

plotFH	<i>Plot the raw data for a FlowHist object</i>
--------	--

Description

Creates a simple plot of the raw histogram data. Used as a utility for other plotting functions, and perhaps useful for users who wish to create their own plotting routines.

Usage

```
plotFH(fh, ...)
```

Arguments

fh	a FlowHist object
...	additional parameters passed to plot

Value

Not applicable, used for plotting

Author(s)

Tyler Smith

Examples

```
library(flowPloidyData)
fh1 <- FlowHist(file = flowPloidyFiles[1], channel = "FL3.INT.LIN")
plotFH(fh1)
```

setBins	<i>setBins</i>
---------	----------------

Description

(Re-) set the bins for a FlowHist object

Usage

```
setBins(fh, bins = 256)
```

Arguments

fh	a FlowHist object
bins	integer, the number of bins to use in aggregating FCS data

Details

This function sets (or resets) the number of bins to use in aggregating FCS data into a histogram, and generates the corresponding data matrix.

The histData matrix also contains the SCvals column. This is used to calculate the single-cut debris component in the NLS model.

Value

a FlowHist object, with the bins slot set to bins, and the corresponding binned data stored in a matrix in the histData slot. Any previous analysis slots are removed: peaks, comps, model, init, nls, counts, CV,

Author(s)

Tyler Smith

Examples

```
## defaults to 256 bins:
library(flowPloidyData)
fh1 <- FlowHist(file = flowPloidyFiles[1], channel = "FL3.INT.LIN")
plot(fh1)
## reset them to 512 bins:
fh1 <- setBins(fh1, 512)
plot(fh1)
```

tabulateFlowHist	<i>exportFlowHist</i>
------------------	-----------------------

Description

Extract analysis results from a flowHist object

Usage

```
tabulateFlowHist(fh, file = NULL)
```

Arguments

fh	a flowHist object, or a list of flowHist objects.
file	character, the name of the file to save data to

Details

A convenience function for extracting the results of the NLS curve-fitting analysis on a flowHist object.

If fh is a single FlowHist object, a data.frame with a single row is returned. If fh is a list of FlowHist objects, a row for each object will be added to the data.frame.

If a file name is provided, the data will be saved to that file.

Value

a data frame

Author(s)

Tyler Smith

Examples

```
library(flowPloidyData)
fh1 <- FlowHist(file = flowPloidyFiles[1], channel = "FL3.INT.LIN")
fh1 <- fhAnalyze(fh1)
tabulateFlowHist(fh1)
```

updateFlowHist	<i>updateFlowHist</i>
----------------	-----------------------

Description

Update, and optionally re-analyze, a FlowHist object

Usage

```
updateFlowHist(fh, linearity = NULL, debris = NULL, analyze = TRUE)
```

Arguments

fh	a FlowHist object
linearity	character, either "variable", the default, or "fixed". If "fixed", linearity is fixed at 2; if "variable", linearity is fit as a model parameter.
debris	character, either "SC", the default, or "MC", to set the debris model component to the Single-Cut or Multi-Cut models.
analyze	boolean, if TRUE the updated model will be analyzed immediately

Details

Allows users to switch the debris model from Single-Cut to Multi-Cut (or vice-versa), or to toggle linearity between fixed and variable.

Value

a FlowHist object with the modified values of linearity and/or debris, and, if analyze was TRUE, a new NLS fitting

Author(s)

Tyler Smith

Examples

```
## defaults to 256 bins:
library(flowPloidyData)
fh1 <- FlowHist(file = flowPloidyFiles[1], channel = "FL3.INT.LIN")
## default is Single-Cut, change that to Multi-Cut:
fh1mc <- updateFlowHist(fh1, debris = "MC")
plot(fh1)
```

<code>viewFlowChannels</code>	<i>viewFlowChannels</i>
-------------------------------	-------------------------

Description

Displays the column names present in an FCS file

Usage

```
viewFlowChannels(file)
```

Arguments

`file` character, the name of a FCS data file

Details

A convenience function for viewing column names in an FCS, in order to select one for the channel argument in `FlowHist`.

Value

A vector of column names from the FCS file.

Author(s)

Tyler Smith

See Also

`FlowHist`

Examples

```
library(flowPloidyData)
viewFlowChannels(flowPloidyFiles[1])
```

Index

`batchFlowHist (FlowHist)`, [5](#)
`browseFlowHist`, [2](#)

`cleanPeaks (findPeaks)`, [3](#)

`fhAnalyze`, [2](#)
`findPeaks`, [3](#)
`FlowHist`, [3](#), [5](#)

`gauss`, [7](#)

`pickInit`, [7](#)
`plot.FlowHist`, [8](#)
`plotFH`, [9](#)

`setBins`, [9](#)

`tabulateFlowHist`, [10](#)

`updateFlowHist`, [11](#)

`viewFlowChannels`, [12](#)