Package 'flowDensity'

November 12, 2026

Type Package

Title Sequential Flow Cytometry Data Gating

Version 1.46.0

Date 2014-10-14

Author Mehrnoush Malek,M. Jafar Taghiyar

Maintainer Mehrnoush Malek <mehrmalek@gmail.com>

Description This package provides tools for automated sequential gating analogous to the manual gating strategy based on the density of the data.

Imports flowCore, graphics, flowViz (>= 1.42), car, polyclip, gplots, methods, stats, grDevices

License Artistic-2.0

biocViews Bioinformatics, FlowCytometry, CellBiology, Clustering, Cancer, FlowCytData, DataRepresentation, StemCell, DensityGating

Suggests knitr,rmarkdown

LazyLoad yes

VignetteBuilder knitr

SystemRequirements xml2, GNU make, C++11

git_url https://git.bioconductor.org/packages/flowDensity

git_branch RELEASE_3_23

git_last_commit d3fbcb7

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

CellPopulation-class . . . 2
deGate . . . 3
flowDensity-methods . . . 5
getflowFrame . . . 6
getPeaks . . . 7
nmRemove . . . 8
notSubFrame . . . 9
plotDens . . . 10

Index **12**

`CellPopulation-class` *Class* `"CellPopulation"`

Description

This class represents the output of 'flowDensity(.)' function from flowDensity package.

Objects from the Class

Objects can be created by calls of the form `new("CellPopulation", ...)`.

Slots

`flow.frame`**:** Object of class `"flowFrame"` representing the flow cytometry data of the cell population

`proportion`**:** Object of class `"numeric"` representing proportion of the cell population with respect to its parent cell population

`cell.count`**:** Object of class `"numeric"` representing cell count of the cell population

`channels`**:** Object of class `"character"` representing channel names corresponding to the 2 dimensions where the cell population is extracted

`position`**:** Object of class `"logical"` representing position of the cell population in the 2-dimensional space

`gates`**:** Object of class `"numeric"` representing thresholds on each channel used to gate the cell population

`filter`**:** Object of class `"matrix"` representing boundary of the cell population using a convex polygon

`index`**:** Object of class `"numeric"` representing indices of the data points in the cell population with respect to its parent cell population

Methods

flowDensity `signature(obj = "CellPopulation", channels = "ANY", position = "logical", singlet.gate = "missing")`: ...

flowDensity `signature(obj = "CellPopulation", channels = "missing", position = "missing", singlet.gate = "logical")`: ...

getflowFrame `signature(obj = "CellPopulation")`: ...

plot `signature(x = "flowFrame", y = "CellPopulation")`: ...

Author(s)

Jafar Taghiyar <email: <jtaghiyar@bccrc.ca»

Examples

```
showClass("CellPopulation")
```

---

deGate *1D density gating method*

Description

Find the best threshold for a single channel in flow cytometry data based on its density distribution.

Usage

```
deGate(obj,channel, n.sd = 1.5, use.percentile = FALSE, percentile =NA,use.upper=FALSE, upper = NA,ver
            bimodal=F,after.peak=NA,alpha = 0.1, sd.threshold = FALSE, all.cuts = FALSE,
            tinypeak.removal=1/25, adjust.dens = 1,count.lim=20,magnitude=.3,slope.w=4,seq.w = 4, spa
```

Arguments

`obj`	obj: a 'FlowFrame' object, 'CellPopulation' or 'GatingHierarchy'
`channel`	a channel's name or its corresponding index in the 'flow.frame'.
`n.sd`	an integer coefficient for the standard deviation to determine the threshold based on the standard deviation if 'sd.threshold' is TRUE.
`use.percentile`	if TRUE, forces to return the 'percentile'th threshold.
`percentile`	A value in [0,1] that is used as the percentile. The default is NA. If set to a value(n) and use.percentile=F, it returns the n-th percentile, for 1-peak populations.
`use.upper`	Logical. If TRUE, forces to return the inflection point based on the first (last) peak if upper=F (upper=T). Default value is set to 'FALSE'
`upper`	if TRUE, finds the change in the slope at the tail of the density curve, if FALSE, finds it at the head. Default value is set to 'NA'.

verbose	Logical. If TRUE, Prints a message if only one peak is found, or when inflection point is used to set the gates.
twin.factor	a value in [0,1] that is used to exclude twinpeaks
bimodal	Logical. If TRUE, it returns a cutoff that splits population closer to 50-50, when there are more than two peaks.
after.peak	Logical. If TRUE, it returns a cutoff that is after the maximum peaks, when there are more than two peaks.
alpha	a value in [0,1) specifying the significance of change in the slope being detected. This is by default 0.1, and typically need not be changed.
sd.threshold	if TRUE, uses 'n.sd' times standard deviation as the threshold. Default value is set to 'FALSE'.
all.cuts	if TRUE, returns all the identified cutoff points, i.e. potential thresholds for that channel. Default value is set to 'FALSE'.
tinypeak.removal	A number in [0,1] to exclude/include tiny peaks in density distribution.
adjust.dens	The smoothness of density in [0,Inf] to be used in density(.). The default value is 1 and should not be changed unless necessary
count.lim	minimum limit for events count in order to calculate the threshold. Default is 20, returning NA as threshold.
magnitude	A value between 0 and 1, for tracking a slope and reporting changes that are smaller than magnitude*peak_height
slope.w	window.width for tracking slope. Default is 4, calculating a slope based on 4 points before and after the current point.
seq.w	value used for making the sequence of density points, used in trackSlope.
spar	value used in smooth.spline function, used in generating the density, default is 0.4.
...	Extra arguments to be passed to smoothSpline function.

Details

deGate works for GatingHierarchy, flowFrame, CellPopulation object or a numeric vector of data. In case the input is a numeric vector, channel doesn't need to provided, but the rest of arguments can be used to tune the outcome.

Value

an integer value (vector) of cutoff(s), i.e. threshold(s), on the specified channel

Author(s)

Mehrnoush Malek «mmalekes@bccrc.ca»

See Also

getflowFrame notSubFrame flowDensity

Examples

```
data_dir <- system.file("extdata", package = "flowDensity")
load(list.files(pattern = 'sampleFCS_1', data_dir, full = TRUE))
#Find the threshold for CD20
cd19.gate <- deGate(f,channel="PerCP-Cy5-5-A")
# Gate out the CD20- populations using the notSubFrame
plotDens(f,c("APC-H7-A","PerCP-Cy5-5-A"))
abline(h=cd19.gate,lty=3,col=2)
```

`flowDensity-methods` *Methods for Function* `flowDensity` *in Package* **flowDensity**

Description

Methods for function `flowDensity` in package **flowDensity**

Arguments

`obj`	GatingHierarchy or `CellPopulation`object
`channels`	a vector of two channel names or their corresponding indices.
`position`	a vector of two logical values specifying the position of the cell subset of interest on the 2D plot.
`...`	This can be used to pass one of the following arguments: • 'use.percentile' if TRUE, returns the 'percentile'th threshold. • 'percentile' a value in [0,1] that is used as the percentile if 'use.percentile' is TRUE. • 'upper' if 'TRUE', it finds the change in the slope after the peak with index 'peak.ind'. • 'use.upper' if 'TRUE', forces to return the inflection point based on the first (last) peak if upper=F (upper=T) • 'twin.factor' a value in [0,1] that is used to exclude twinpeaks. • 'bimodal' If TRUE, it returns a cutoff that splits population closer to 50-50, when there are more than two peaks. • 'after.peak' If TRUE, it returns a cutoff that is after the maximum peaks, when there are more than two peaks. • 'sd.threshold' if TRUE, it uses 'n.sd' times standard deviation for gating. • 'n.sd' an integer that is multiplied to the standard deviation to determine the place of threshold if 'sd.threshold' is 'TRUE'. • 'tinypeak.removal' a vector of length 2, for sensitivity of peak finding for each channel. See deGate() for more information. • 'filter' If provided it uses the given filter to gate the population. • 'use.control' if TRUE, it finds the threshold using a matched control population and uses it for gating.

- 'control' a 'flowFrame' or 'CellPopulation' object used for calculating the gating threshold when 'use.control' is set to TRUE. If a control population is used, the other arguments ('upper', 'percentile', etc.) are applied to the control data when finding the threshold (i.e. not to 'obj').
- 'alpha' a value in [0,1) specifying the significance of change in the slope which would be detected. This is by default 0.1, and typically need not be changed.
- 'ellip.gate' if TRUE, it fits an ellipse on the data as a gate, otherwise the rectangle gating results are returned
- 'scale' a value in [0,1) that scales the size of ellipse to fit if 'ellip.gate' is TRUE

Value

a CellPopulation object.

`getflowFrame` *'CellPopulation' class accessor.*

Description

an accessor for 'CellPopulation' class to get its 'FlowFrame' object. This will remove all the NA values in the frame.

Usage

```
getflowFrame(obj)
```

Arguments

`obj`	a 'CellPopulation' object.

Value

a 'FlowFrame' object.

Author(s)

Jafar Taghiyar «`jtaghiyar@bccrc.ca`»

Examples

```
data_dir <- system.file("extdata", package = "flowDensity")
load(list.files(pattern = 'sampleFCS_1', data_dir, full = TRUE))
lymph <- flowDensity(obj=f, channels=c('FSC-A', 'SSC-A'),
                     position=c(TRUE, NA))
f.lymph <- getflowFrame(lymph)
```

getPeaks *Finding Peaks*

Description

Find all peaks in density along with their indices

Usage

```
getPeaks(obj, channel,tinypeak.removal=1/25, adjust.dens=1,verbose=F,twin.factor=1,spar = 0.4,...)
```

Arguments

obj	a 'FlowFrame', 'GatingHierarchy', 'CellPopulation' a density object or a numeric vector of density.
channel	a channel's name or its corresponding index. If the input is numeric vector, channel is NA.
tinypeak.removal	A number in [0,1] to exclude/include tiny peaks in density distribution. Default is 1/25.
adjust.dens	The smoothness of density in [0,Inf] to be used in density(.). The default value is 1 and should not be changed unless necessary
verbose	If TRUE, printing warnings.
twin.factor	If smaller than 1, peaks that are of greater than hieght as the maximum peak*twin.factor will be removed.
spar	argument to pass to smoothSpline function, default value of spar is 0.4.
...	Other arguments that can be passed to smoothSpline function.

Value

a list, including peaks, their corresponding indices and height.

Author(s)

Mehrnoush Malek «mmalekes@bccrc.ca»

See Also

deGate notSubFrame flowDensity

Examples

```
data_dir <- system.file("extdata", package = "flowDensity")
load(list.files(pattern = 'sampleFCS_1', data_dir, full = TRUE))
#Find the threshold for CD20
peaks <- getPeaks(f,channel="PerCP-Cy5-5-A",tinypeak.removal=1/30)
peaks
```

nmRemove *Preprocessing helper function for flow cytometry data*

Description

Remove the margin events on the axes. Usually, these events are considered as debris or artifacts. This is specifically useful for 'FSC' and 'SSC' channels in a 'FlowFrame' object. However, any channel can be input as an argument.

Usage

```
nmRemove(  flow.frame, channels, neg=FALSE, verbose=FALSE,return.ind=FALSE)
```

Arguments

flow.frame	a 'FlowFrame' object.
channels	a vector of channel names or their corresponding indices.
neg	if TRUE, negative events are also removed
verbose	if TRUE, it prints the margin event in each channel
return.ind	if TRUE, it return indices of margin events for each channel.

Value

a 'FlowFrame' object, or a 'list' of indices identifying margin events for each channel.

Author(s)

Jafar Taghiyar «jtaghiyar@bccrc.ca» Mehrnoush Malek «mmalekes@bccrc.ca»

Examples

```
data_dir <- system.file("extdata", package = "flowDensity")
load(list.files(pattern = 'sampleFCS_2', data_dir, full = TRUE))
#Removing margin events of FSC-A and SSC-A channels
no.margin <- nmRemove(f2, c("FSC-A","SSC-A"),verbose=TRUE)
plotDens(f2, c("FSC-A","SSC-A"))
# Scatter plot of FSC-A vs. SSC-A after removing margins
plotDens(no.margin, c("FSC-A","SSC-A"))
```

notSubFrame *Removing a subset of a FlowFrame object*

Description

Remove a subset of a FlowFrame object specified by gates from the flowDensity method. It comes in handy when one needs the complement of a cell population in the input flow cytometry data.

Usage

```
notSubFrame(obj, channels, position = NA, gates, filter)
```

Arguments

obj	a 'FlowFrame' or 'cellPopulation' object.
channels	a vector of two channel names or their corresponding indices in the 'flow.frame'.
position	a vector of two logical values specifying the position of the cell subset of interest on the 2D plot.
gates	the gates slot in the CellPoulation object which is output by flowDensity function. It can also be a vector of two integer values each of which specifies a threshold for the corresponding channel in 'channels' argument.
filter	boundary of the subset to be removed. This value is stored in the 'filter' slot of a 'CellPopulation' object.

Value

a CellPopulation object.

Author(s)

Mehrnoush Malek «mmalekes@bccrc.ca»

Examples

```
data_dir <- system.file("extdata", package = "flowDensity")
load(list.files(pattern = 'sampleFCS_1', data_dir, full = TRUE))
#Find the threshold for CD20
cd20.gate <- deGate(f,channel="APC-H7-A")
# Gate out the CD20- populations using the notSubFrame
CD20.pos <- notSubFrame(f,channels=c("APC-H7-A","PerCP-Cy5-5-A"),position=c(FALSE,NA),gates=c(cd20.gate,NA))
#Plot the CD20+ cells on same channels
plotDens(CD20.pos@flow.frame,c("APC-H7-A","PerCP-Cy5-5-A"))
```

plotDens *Plot flow cytometry data with density-based colors*

Description

Generate a scatter dot plot with colors based on the distribution of the density of the provided channels.

Usage

```
plotDens(obj, channels ,col, main, xlab, ylab, xlim,ylim, pch=".", density.overlay=c(FALSE,FALSE),coun
dens.type=c("l","l"),transparency=1, adjust.dens=1,show.contour=F, contour.col="darkgrey", verbose=T
```

Arguments

obj	a 'FlowFrame', or 'cellPopulation' object.
channels	a vector of two channel names or their corresponding indices in the 'flow.frame'.
col	A specification for the default plotting color: see '?par'.
main	an overall title for the plot: see '?plot'
xlab	a title for the x axis: see '?plot'
ylab	a title for the y axis: see '?plot'
xlim	a range for the x axis: see '?plot'
ylim	a range for the y axis: see '?plot'
pch	Either an integer specifying a symbol or a single character to be used as the default in plotting points: see '?par'.
density.overlay	Logical vector of length 2, to plot density overlays on the x and y axes. Default is c(FALSE,FALSE).
count.lim	Cutoff for number of events to set color. Default is 20. Samples with less than 20 cells will be plotted in black.
dens.col	2-character string giving the color of plot desired for density curves.
cex	Size of the points for the plot. For more information look at ?plot in graphics.
dens.type	2-character string giving the type of plot desired.
transparency	Transparency of the bi-variate plot, to see the densitu curves in the background. The lower it is, the more transparent the plot is.
adjust.dens	The smoothness of density in [0,Inf] to be used in density(.). The default value is 1 and should not be changed unless necessary
show.contour	Default is FALSE. It add the contourLines to plot.
contour.col	Color for contourLines. Default is darkgrey.
verbose	Default is True. It will add that the sample has 0 cells in the plot title.
...	can be used to provide desired arguments for the plot() function used to plot the output results.

Value

a scatter dot plot with density-based colors, along with density overlays if desired. Set xlim and ylim when plotting if you would like to have all your plots to have same range on the axes (specially when density.overlay=TRUE)

Author(s)

Mehrnoush Malek «mmalekes@bccrc.ca» Jafar Taghiyar «jtaghiyar@bccrc.ca»

Examples

```
data_dir <- system.file("extdata", package = "flowDensity")
load(list.files(pattern = 'sampleFCS_1', data_dir, full = TRUE))
#Plot CD3 vs. CD19 to see the distribution of cell populations and their density
plotDens(f,c("V450-A","PerCP-Cy5-5-A"))
```

Index

∗ **Automated gating**
plotDens, 10
∗ **FlowCytData**
plotDens, 10
∗ **classes**
CellPopulation-class, 2

CellPopulation, *5*
CellPopulation-class, 2

deGate, 3, *7*

flowDensity, *4*, *7*
flowDensity,CellPopulation,ANY,logical
*(*flowDensity-methods*)*, 5
flowDensity,flowFrame,ANY,logical
*(*flowDensity-methods*)*, 5
flowDensity-methods, 5

getflowFrame, *4*, 6
getflowFrame, CellPopulation-method
*(*getflowFrame*)*, 6
getflowFrame,CellPopulation-method
*(*CellPopulation-class*)*, 2
getPeaks, 7

nmRemove, 8
notSubFrame, *4*, *7*, 9

plot, flowFrame, CellPopulation-method
*(*plotDens*)*, 10
plot,flowFrame,CellPopulation-method
*(*CellPopulation-class*)*, 2
plotDens, 10