Package 'flowCut'

November 12, 2026

Type Package

Title Automated Removal of Outlier Events and Flagging of Files Based on Time Versus Fluorescence Analysis

Version 1.22.0

Date 2022-03-24

Depends R (>= 3.4), flowCore

Imports flowDensity (>= 1.13.1), Cairo, e1071, grDevices, graphics, stats,methods

Description Common techinical complications such as clogging can result in spurious events and fluorescence intensity shifting,
flowCut is designed to detect and remove technical artifacts from your data by removing segments that
show statistical differences from other segments.

VignetteBuilder knitr

Suggests RUnit, BiocGenerics, knitr, markdown, rmarkdown

biocViews FlowCytometry, Preprocessing, QualityControl, CellBasedAssays

License Artistic-2.0

LazyLoad yes

RoxygenNote 6.0.1

NeedsCompilation no

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

git_branch RELEASE_3_23

git_last_commit 80c33b6

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Justin Meskas [cre, aut], Sherrie Wang [aut]

Maintainer Justin Meskas <justinmeskas@gmail.com>

Contents

flowCut 2
flowCutData 6
removeLowDensSections 7

Index **9**

`flowCut` *Precise and Accurate Automated Removal of Outlier Events and Flagging of Files Based on Time Versus Fluorescence Analysis.*

Description

flowCut automatically removes outlier events in flow cytometry data files due to abnormal flow resulting from clogs and other common technical problems.

Usage

```
flowCut (f,
        Segment=500,
        Channels=NULL,
        Directory=NULL,
        FileID=NULL,
        Plot="Flagged Only",
        MaxContin=0.1,
        MeanOfMeans=0.13,
        MaxOfMeans=0.15,
        MaxValleyHgt=0.1,
        MaxPercCut=0.3,
        LowDensityRemoval=0.1,
        GateLineForce=NULL,
        UseOnlyWorstChannels=FALSE,
        AmountMeanRangeKeep=1,
        AmountMeanSDKeep=2,
        PrintToConsole=FALSE,
        AllowFlaggedRerun=TRUE,
        UseCairo=FALSE,
        UnifTimeCheck=0.22,
        RemoveMultiSD=7,
        AlwaysClean=FALSE,
        IgnoreMonotonic=FALSE,
        MonotonicFix=NULL,
        Measures=c(1:8),
        Verbose=FALSE)
```

Arguments

f	A single flowFrame to be processed.
Segment	An integer value that specifies the number of events in each segment to be analyzed. A single segment is defaulted to contain 500 events.
Channels	A vector of indices for the channels to be cleaned. The default is NULL which will clean all non scatter and non time channels.
Directory	A directory that specifies where system output files are stored. If NULL, a directory will be created automatically under the current working directory.
FileID	A character or numeric variable of a unique ID for the file. If NULL, FileID will be randomly generated.
Plot	A character variable that indicates the amount of flowFrame plots that will be generated. The default is "Flagged Only". Other values are "All" and "None". "All" will generate plots for both passed and flagged files, "Flagged Only" only generates plots for files that display at least one of the four flagging behaviours, where as "None" will produce no figures.
MaxContin	A numeric value that defines the critical value at which the difference of the mean of a given segment to the mean of an adjacent segment divided by the difference of the 98th percentile and the 2nd percentile for the whole file becomes significant, indicating a sudden change in the mean for neighbouring segments. The value is defaulted at 0.1.
MeanOfMeans	A numeric value that defines the critical mean value of percentage differences between the difference of the maximum and minimum mean and the difference of the 98th and 2nd percentile from all segments together. If the value exceeds the defaulted critical value of 0.13, which indicates a large gradual change of fluorescence in all channels, then the file is flagged. Formula: mean((max(z)-min(z)) / (98th percentile - 2nd percentile)) >= MeanOfMeans, where z is all the means of the segments.
MaxOfMeans	A numeric value that defines the critical maximum value of percentage differences between the difference of the maximum and minimum mean and the difference of the 98th and 2nd percentile from all segments together. If the value exceeds the defaulted critical value of 0.15, which indicates a very large gradual change of fluorescence in one channel, then the file is flagged. Formula: max((max(z)-min(z)) / (98th percentile - 2nd percentile)) >= MaxOfMeans, where z is all the means of the segments.
MaxValleyHgt	A numeric value that defines the maximum height of the intersection point of the cutoff line and the density of summed measure plot. Increasing MaxValleyHgt will potentially increase the amount of events being cut. The default is 0.1.
MaxPercCut	A numeric value between 0 to 1 that sets the percentage of events that a user is willing to potentially cut. If the file contains a lot of bad events, we suggest increasing this number to potentially cut off more deviates; usually 0.5 is sufficient in this case. The default is 0.3.
LowDensityRemoval	A numerical value between 0 and 1. Any events having a density of less than LowDensityRemoval are removed. The default is 0.1.

`GateLineForce` A numeric value that forces the line in the density distribution plot to a particular value. This will be useful if the user runs a file once and wants to see what the cleaning process would look like if a little more or less was removed. The default is NULL.

`UseOnlyWorstChannels`
A logical value that allows for an automated detection of the worst channels. These channels will be the only ones used for cleaning. Focusing on the worst channels can often make the results more consistent with intuition. The default is FALSE.

`AmountMeanRangeKeep`
A non negative integer value (including 0) that defines the number of channels with a large range of means that would be included if UseOnlyWorstChannels is set to TRUE. The default is 1.

`AmountMeanSDKeep`
A non negative integer value (including 0) that defines the number of channels with a large standard deviation of means that would be included if UseOnlyWorstChannels is set to TRUE. The default is 2.

`PrintToConsole` A logical value that dictates if the figure is printed to the console or in the current directory. If no figure is desired, please see the Plot argument. The default is FALSE.

`AllowFlaggedRerun`
A logical value that dictates if flowCut will run a second time. However, this only occurs if the file was flagged on the first run. The default is TRUE.

`UseCairo` A logical value that dictates if the Cairo package should be used to save the png file. If FALSE, png() in grDevices will be used. Note that some operating systems might only work with one or the other. Default is FALSE.

`UnifTimeCheck` A numeric value for the cutoff point when flowCut decides to run or not run its code. Having too high of a value will increase the chances that flowCut crashes. This issue occurs when the time channel is not uniform. A simple test is done: If the mean of the time marker is larger than UnifTimeCheck away from the mid-point then the file is not run. Default is set at 0.22. This is a proportion of the range, and hence must be between 0 and 0.5. Consider only to change this slightly more positive to allow a few files to have cleaning if they are borderline cases.

`RemoveMultiSD` A numeric value used to remove very statistically different segments even when the file is nice. It is the amount of standard deviations away from the mean of all segments of which anything larger will be removed. Default is 7, not advised to reduce this below 3.

`AlwaysClean` A logical value to clean the file (or not) with the multi standard deviation removal method (See RemoveMultiSD). If it passes all the tests (MeanOfMeans, MaxOfMeans and MaxContin) it is deemed clean enough and will skip further cleaning if AlwaysClean = FALSE. Otherwise, AlwaysClean = TRUE, the code will attempt to clean the file without the requirement of being flagged by one of the tests (MeanOfMeans, MaxOfMeans and MaxContin).

`IgnoreMonotonic`
A logical value used to ignore the monontonically increasing in time flagging test. If TRUE it will ignore the test. Default is FALSE.

`MonotonicFix` A numeric value greater or equal to 0 used to fix the monontonic issues in time. If NULL, the default, the fix is skipped. The value is how large of a jump in backwards time that is allowed, anything larger will be corrected. Suggestion to use 1. Set to 0 to fix any monotonic in time issues for the entire file. See figures inside the folder "/Mono/" which is inside "Directory" (or your current directory if "Directory" is NULL). Plots will only be plotted if "Plot=All" or "Plot=Flagged Only".

`Measures` A numeric vector that determines which measures are used when calculating which segments are statistically different. The default is c(1:8). Try any subset of c(1:8) for the desired subset of 5th and 20th percentiles, median, 80th and 95th percentiles, mean, variance and skewness, respectively.

`Verbose` A logical value that dictates if computational information is printed while the code is running. The default is FALSE.

Details

flowCut's methodology is based on identifying both regions of low density and segments (default size of 500 events) that are significantly different from the rest. Eight measures of each segment (mean, median, 5th, 20th, 80th and 95th percentile, second moment (variation) and third moment (skewness)) are calculated. The density of the summation of the 8 measures and two parameters (MaxValleyHgt and MaxPercCut) will determine which events are significantly different and these will be removed. We also flag files if they display any of the following: 1) not monotonically increasing in time, 2) sudden changes in fluorescence, 3) large gradual change of fluorescence in all channels or 4) very large gradual change of fluorescence in one channel.

Value

A list containing four elements. The first element, $frame, is the flowFrame file returned after flowCut has cleaned the flowFrame. The second element, $ind, is a vector containing indices of events being removed. The third element, $data, is a table containing computational information that users can access. Information includes: "Is it monotonically increasing in time", "Largest continuous jump", "Continuous - Pass", "Mean of percentage of range of means divided by range of data", "Mean of % - Pass", "Max of % of range of means divided by range of data", "Max of % - Pass", "Has a low density section been removed", "% of low density removed", "How many Segments have been removed", "% of events removed from Segments removed", "Worst Channel", "% of events removed", "FileID", and "Has the file passed". The fourth element, $worstChan, is the index of the channel with the most drift over time before cleaning.

Author(s)

Maintainer: Justin Meskas <justinmeskas@gmail.com>, Sherrie Wang <swang@bccrc.ca>

Examples

```
data(flowCutData)
res_flowCut <- flowCut(flowCutData[[1]], Plot="All")
# See plot in working directory/flowCut

# Here is an example with decreasing the MaxValleyHgt parameter, which causes a flag.
```

```
res_flowCut <- flowCut(flowCutData[[2]], MaxValleyHgt = 0.01, Verbose = TRUE, AllowFlaggedRerun = FALSE, Plot="All
res_flowCut$data

# Compare with:
res_flowCut <- flowCut(flowCutData[[2]], Verbose = TRUE,  AllowFlaggedRerun = FALSE, Plot="All")

# Here is an example with decreasing the MaxPercCut parameter, which causes a flag.
res_flowCut <- flowCut(flowCutData[[3]], MaxPercCut = 0.15,  AllowFlaggedRerun = FALSE, Verbose = TRUE)
res_flowCut$data

# Compare with:
res_flowCut <- flowCut(flowCutData[[3]], Verbose = TRUE, Plot="All")

# Here is an example using AllowFlaggedRerun and UseOnlyWorstChannels
res_flowCut <- flowCut(flowCutData[[4]], AllowFlaggedRerun = TRUE,
                    UseOnlyWorstChannels = TRUE, Plot="All")
res_flowCut$data

# Alternative plotting option 1
res_flowCut <- flowCut(flowCutData[[1]])
library(flowDensity)
plotDens(flowCutData[[1]], c("Time","FL1-H"))
points(exprs(flowCutData[[1]])[res_flowCut$ind, c("Time","FL1-H")], pch=".")

# Alternative plotting option 2
res_flowCut <- flowCut(flowCutData[[1]], Plot="All", PrintToConsole=TRUE)
```

`flowCutData` *A list containing two GvHD flow frames from flowCore*

Description

We are taking two flowFrames from flowCore's GvHD dataset and two flowFrames from FlowRespository, an open source cytometry data repository. The first two flowFrames are GvHD flowFrame objects 's9a06' and 's5a07'. The third flowFrame is from FlowRespository's FR-FCM-ZZ7E 'Macrophages + oATP.fcs', and the fourth is from FlowRespository's 'FR-FCM-ZZVB Specimen_001_B6 LSK.fcs'. We used the estimateLogical transformation on the data. We then processed the data with flowCut in the examples.

Usage

```
data(flowCutData)
```

Author(s)

Maintainer: Justin Meskas <justinmeskas@gmail.com>, Sherrie Wang <swang@bccrc.ca>

Examples

```
data(flowCutData)
res_flowCut <- flowCut(flowCutData[[1]], Plot="All")
# See plot in working directory/flowCut

# Here is an example with decreasing the MaxValleyHgt parameter, which causes a flag.
res_flowCut <- flowCut(flowCutData[[2]], MaxValleyHgt = 0.01, Verbose = TRUE, AllowFlaggedRerun = FALSE, Plot="All
res_flowCut$data

# Compare with:
res_flowCut <- flowCut(flowCutData[[2]], Verbose = TRUE, AllowFlaggedRerun = FALSE, Plot="All")

# Here is an example with decreasing the MaxPercCut parameter, which causes a flag.
res_flowCut <- flowCut(flowCutData[[3]], MaxPercCut = 0.15, AllowFlaggedRerun = FALSE, Verbose = TRUE)
res_flowCut$data

# Compare with:
res_flowCut <- flowCut(flowCutData[[3]], Verbose = TRUE, Plot="All")

# Here is an example using AllowFlaggedRerun and UseOnlyWorstChannels
res_flowCut <- flowCut(flowCutData[[4]], AllowFlaggedRerun = TRUE,
                    UseOnlyWorstChannels = TRUE, Plot="All")
res_flowCut$data

# Alternative plotting option 1
res_flowCut <- flowCut(flowCutData[[1]])
library(flowDensity)
plotDens(flowCutData[[1]], c("Time","FL1-H"))
points(exprs(flowCutData[[1]])[res_flowCut$ind, c("Time","FL1-H")], pch=".")

# Alternative plotting option 2
res_flowCut <- flowCut(flowCutData[[1]], Plot="All", PrintToConsole=TRUE)
```

---

removeLowDensSections *Remove low density regions of a flowFrame.*

Description

Remove low density regions of a flowFrame.

Usage

```
removeLowDensSections (f, Time.loc, Segment=500, LowDensityRemoval=0.1, Verbose=FALSE)
```

Arguments

f	A single flowFrame to be processed.
Time.loc	An positive interger value for the location of the time channel.

`Segment` An integer value that specifies the number of events in each segment to be analyzed. A single segment is defaulted to contain 500 events.

`LowDensityRemoval` An numeric value between 0 and 1 that specifies the proportion cut-off of the low density region. All events with less density than LowDensityRemoval will be removed. The default is 0.1.

`Verbose` A logical value that dictates if computational information is printed while the code is running. The default is FALSE.

Details

removeLowDensSections removes low density regions of a flowFlame. The flowFrame must have a time channel.

Value

A list containing two elements. The first element, $frame, is the flowFrame file returned after removeLowDensSections has removed the low density sections. The second element, $rem.ind, is a vector containing indices of events being removed.

Author(s)

Maintainer: Justin Meskas <justinmeskas@gmail.com>, Sherrie Wang <swang@bccrc.ca>

Examples

```
data(flowCutData)
res_lowDens <- removeLowDensSections(flowCutData[[1]], LowDensityRemoval=0.1, Time.loc=which(parameters(flowCutD
library(flowDensity)
plotDens(flowCutData[[1]], c("Time","FL1-H"))
points(exprs(flowCutData[[1]])[res_lowDens$rem.ind, c("Time","FL1-H")],
    pch=".", col="grey")
```

Index

∗ **FlowCytData**
flowCut, 2
removeLowDensSections, 7
∗ **datasets, FlowCytData**
flowCutData, 6

flowCut, 2
flowCutData, 6

removeLowDensSections, 7