Package ‘PeacoQC’

October 14, 2021

Title Peak-based selection of high quality cytometry data

Version 1.2.0

Description

This is a package that includes pre-processing and quality control functions that can remove margin events, compensate and transform the data and that will use PeacoQCSignalStability for quality control. This last function will first detect peaks in each channel of the flowframe. It will remove anomalies based on the IsolationTree function and the MAD outlier detection method. This package can be used for both flow- and mass cytometry data.

Encoding UTF-8

License GPL (>=3)

LazyData true

URL http://github.com/saeyslab/PeacoQC

BugReports http://github.com/saeyslab/PeacoQC/issues

Depends R (>= 4.0)

Imports circlize, ComplexHeatmap, flowCore, flowWorkspace, ggplot2, grDevices, grid, gridExtra, methods, plyr, stats, utils

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

biocViews FlowCytometry, QualityControl, Preprocessing, PeakDetection

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

git_branch RELEASE_3_13

git_last_commit 0069897

git_last_commit_date 2021-05-19

Date/Publication 2021-10-14

Author Annelies Emmaneel [aut, cre]

Maintainer Annelies Emmaneel <annelies.emmaneel@hotmail.com>

R topics documented:

PeacoQC *Peak-based detection of high quality cytometry data*

Description

PeacoQC will determine peaks on the channels in the flowframe. Then it will remove anomalies caused by e.g. clogs, changes in speed etc. by using an IsolationTree and/or the MAD method.

Usage

```
PeacoQC(ff, channels, determine_good_cells="all",
        plot=20, save_fcs=TRUE, output_directory=".",
        name_directory="PeacoQC_results", report=TRUE,
        events_per_bin=FindEventsPerBin(remove_zeros, ff, channels,
        min_cells, max_bins, step), min_cells=150, max_bins=500, step=500,
        MAD=6, IT_limit=0.6, consecutive_bins=5, remove_zeros=FALSE,
        suffix_fcs="_QC", force_IT=150, ...)
```

Arguments

ff	A flowframe or the location of an fcs file. Make sure that the flowframe is compensated and transformed. If it is mass cytometry data, only a transformation is necessary.
channels	Indices or names of the channels in the flowframe on which peaks have to be determined.
determine_good_cells	If set to FALSE, the algorithm will only determine peaks. If it is set to "all", the bad measurements will be filtered out based on the MAD and IT analysis. It can also be put to "MAD" or "IT" to only use one method of filtering.
plot	When PeacoQC removes more than the specified percentage, an overview plot will be made of all the selected channels and the deleted measurements. If set to TRUE, the PlotPeacoQC function is run to make an overview plot of the deleted measurements, even when nothing is removed. Default is set to 20. If an increasing or decreasing trend is found, a figure will also be made except if plot is set to FALSE.

`save_fcs` If set to TRUE, the cleaned fcs file will be saved in the `output_directory` as: filename_QC.fcs. The _QC name can be altered with the `suffix_fcs` parameter. An extra column named "Original_ID" is added to this fcs file where the cells are given their original cell id. Default is TRUE.

`output_directory` Directory where a new folder will be created that consists of the generated fcs files, plots and report. If set to NULL, nothing will be stored.The default folder is the working directory.

`name_directory` Name of folder that will be generated in `output_directory`. The default is "PeacoQC_results".

`report` Overview text report that is generated after PeacoQC is run. If set to FALSE, no report will be generated. The default is TRUE.

`events_per_bin` Number of events that are put in one bin. Default is calculated based on the rows in `ff`

`min_cells` The minimum amount of cells (nonzero values) that should be present in one bin. Lowering this parameter can affect the robustness of the peak detection. Default is 150.

`max_bins` The maximum number of bins that can be used in the cleaning process. If this value is lowered, larger bins will be made. Default is 500.

`step` The step in events_per_bin to which the parameter is reduced to. Default is 500.

`MAD` The MAD parameter. Default is 6. If this is increased, the algorithm becomes less strict.

`IT_limit` The IsolationTree parameter. Default is 0.55. If this is increased, the algorithm becomes less strict.

`consecutive_bins` If 'good' bins are located between bins that are removed, they will also be marked as 'bad'. The default is 5.

`remove_zeros` If this is set to TRUE, the zero values will be removed before the peak detection step. They will not be indicated as 'bad' value. This is recommended when cleaning mass cytometry data. Default is FALSE.

`suffix_fcs` The suffix given to the new fcs files. Default is "_QC".

`force_IT` If the number of determined bins is less than this number, the IT analysis will not be performed. Default is 150 bins.

`...` Options to pass on to the `PlotPeacoQC` function (display_cells, manual_cells, prefix)

Value

This function returns a `list` with a number of items. It will include "FinalFF" where the transformed, compensated and cleaned flowframe is stored. It also contains the starting parameters and the information necessary to give to `PlotPeacoQC` if the two functions are run seperatly. The GoodCells list is also given where 'good' measurements are indicated as TRUE and the to be removed measurements as FALSE.

Examples

```
# General pipeline for preprocessing and quality control with PeacoQC

# Read in raw fcs file
fileName <- system.file("extdata", "111.fcs", package="PeacoQC")
ff <- flowCore::read.FCS(fileName)

# Define channels where the margin events should be removed
# and on which the quality control should be done
channels <- c(1, 3, 5:14, 18, 21)

ff <- RemoveMargins(ff=ff, channels=channels, output="frame")

# Compensate and transform the data

ff <- flowCore::compensate(ff, flowCore::keyword(ff)$SPILL)
ff <- flowCore::transform(ff,
                            flowCore::estimateLogicle(ff,
                            colnames(flowCore::keyword(ff)$SPILL)))
#Run PeacoQC
PeacoQC_res <- PeacoQC(ff, channels,
                        determine_good_cells="all",
                        save_fcs=TRUE)
```

PeacoQCHeatmap *Make overview heatmap of quality control analysis*

Description

PeacoQCHeatmap will make a heatmap to display all the results generated by PeacoQC. It will include the percentages of measurements that are removed in total, by the IT method and by the MAD method. It will also show the parameters that were used during the quality control.

Usage

```
PeacoQCHeatmap(report_location, show_values=TRUE, show_row_names=TRUE,
latest_tests=FALSE, title="PeacoQC report", ...)
```

Arguments

report_location
: The path to the PeacoQC report generated by PeacoQC.

show_values
: If set to TRUE, the percentages of removed values will be displayed on the heatmap. Default is TRUE.

show_row_names
: If set to FALSE, the filenames will not be displayed on the heatmap. Default is TRUE.

`latest_tests` If this is set to TRUE, only the latest quality control run will be displayed in the heatmap. Default is FALSE.

`title` The title that should be given to the heatmap. Default is "PeacoQC_report".

`...` Extra parameters to be given to the `Heatmap` function (eg. row_split)

Value

This function returns nothing but generates a heatmap that can be saved as pdf or png

Examples

```
# Find path to PeacoQC report
location <- system.file("extdata", "PeacoQC_report.txt", package="PeacoQC")

# Make heatmap overview of quality control run
PeacoQCHeatmap(report_location=location)

# Make heatmap with only the runs of the last test
PeacoQCHeatmap(report_location=location, latest_tests=TRUE)

# Make heatmap with row annotation
PeacoQCHeatmap(report_location=location,
              row_split=c(rep("r1",7), rep("r2", 55)))
```

`PlotPeacoQC` *Visualise deleted cells of PeacoQC*

Description

`PlotPeacoQC` will generate a png file with on overview of the flow rate and the different selected channels. These will be annotated based on the measurements that were removed by PeacoQC. It is also possible to only display the quantiles and median or only the measurements without any annotation.

Usage

```
PlotPeacoQC(ff, channels, output_directory=".", display_cells=2000,
            manual_cells=NULL, title_FR=NULL, display_peaks=TRUE,
            prefix="PeacoQC_", time_unit=100, ...)
```

Arguments

`ff` A flowframe

`channels` Indices of names of the channels in the flowframe that have to be displayed

`output_directory`
Directory where the plots should be generated. Set to NULL if no plots need to be generated. The default is the working directory.

`display_cells` The number of measurements that should be displayed. (The number of dots that are displayed for every channel) The default is 5000.

`manual_cells` Give a vector (TRUE/FALSE) with annotations for each cell to compare the automated QC with. The default is NULL.

`title_FR` The title that has to be displayed above the flow rate figure. Default is NULL.

`display_peaks` If the result of `PeacoQC` is given, all the quality control results will be visualised. If set to TRUE: `PeacoQC` will be run and only the peaks will be displayed without any quality control. If set to FALSE, no peaks will be displayed and only the events will be displayed. Default is TRUE.

`prefix` The prefix that will be given to the generated png file. Default is "PeacoQC_".

`time_unit` The number of time units grouped together for visualising event rate. The default is set to 100, resulting in events per second for most flow datasets. Suggested to adapt for mass cytometry data.

`...` Arguments to be given to `PeacoQC` if `display_peaks` is set to TRUE.

Value

This function returns nothing but generates a png file in the output_directory

Examples

```
## Plotting the results of PeacoQC

# Read in transformed and compensated data
fileName <- system.file("extdata", "111_Comp_Trans.fcs", package="PeacoQC")
ff <- flowCore::read.FCS(fileName)

# Define channels on which the quality control should be done and the
# plots should be made
channels <- c(1, 3, 5:14, 18, 21)

# Run PeacoQC
PeacoQC_res <- PeacoQC(ff,
    channels,
    determine_good_cells="all",
    plot=FALSE,
    save_fcs=TRUE)

# Run PlotPeacoQC
PlotPeacoQC(ff, channels, display_peaks=PeacoQC_res)

## Plot only the peaks (No quality control)
PlotPeacoQC(ff, channels, display_peaks=TRUE)

## Plot only the dots of the file
PlotPeacoQC(ff, channels, display_peaks=FALSE)
```

`RemoveDoublets` *Remove doublet events from flow cytometry data*

Description

`RemoveDoublets` will remove doublet events from the flowframe based on two channels.

Usage

```
RemoveDoublets(ff, channel1="FSC-A", channel2="FSC-H", nmad=4,
verbose=FALSE, output="frame")
```

Arguments

`ff`	A flowframe that contains flow cytometry data.
`channel1`	The first channels that will be used to determine the doublet events. Default is "FSC-A"
`channel2`	The second channels that will be used to determine the doublet events. Default is "FSC-H"
`nmad`	Bandwidth above the ratio allowed (cells are kept if their ratio is smaller than the median ratio + `nmad` times the median absolute deviation of the ratios). Default is 4.
`verbose`	If set to TRUE, the median ratio and width will be printed. Default is FALSE.
`output`	If set to "full", a list with the filtered flowframe and the indices of the doublet event is returned. If set to "frame", only the filtered flowframe is returned. The default is "frame".

Value

This function returns either a filtered flowframe when the `output` parameter is set to "frame" or a list containing the filtered flowframe and a TRUE/FALSE list indicating the margin events. An extra column named "Original_ID" is added to the flowframe where the cells are given their original cell id.

Examples

```
# Read in data
fileName <- system.file("extdata", "111.fcs", package="PeacoQC")
ff <- flowCore::read.FCS(fileName)

# Remove doublets
ff_cleaned <- RemoveDoublets(ff)
```

RemoveMargins *Remove margin events of flow cytometry data*

Description

RemoveMargins will remove margin events from the flowframe based on the internal description of the fcs file.

Usage

```
RemoveMargins(ff, channels,
channel_specifications=NULL, output="frame")
```

Arguments

`ff` A flowframe that contains flow cytometry data.

`channels` The channel indices or channel names that have to be checked for margin events

`channel_specifications`
A list of lists with parameter specifications for certain channels. This parameter should only be used if the values in the internal parameters description is too strict or wrong for a number or all channels. This should be one list per channel with first a minRange and then a maxRange value. This list should have the channel name found back in `colnames(flowCore::exprs(ff))`. If a channel is not listed in this parameter, its default internal values will be used. The default of this parameter is NULL.

`output` If set to "full", a list with the filtered flowframe and the indices of the margin event is returned. If set to "frame", only the filtered flowframe is returned. The default is "frame".

Value

This function returns either a filtered flowframe when the `output` parameter is set to "frame" or a list containing the filtered flowframe and a TRUE/FALSE list indicating the margin events. An extra column named "Original_ID" is added to the flowframe where the cells are given their original cell id.

Examples

```
# Read in raw data
fileName <- system.file("extdata", "111.fcs", package="PeacoQC")
ff <- flowCore::read.FCS(fileName)

# Define channels where the margin events should be removed
channels <- c(1, 3, 5:14, 18, 21)

# Remove margins
```

```
ff_cleaned <- RemoveMargins(ff, channels)

# If an internal value is wrong for a channels (e.g. FSC-A)

channel_specifications <- list("FSC-A"=c(-111, 262144))
ff_cleaned <- RemoveMargins(
    ff,
    channels,
    channel_specifications=channel_specifications)
```

Index