plateCore

March 24, 2012

`Subset` *Subset*

Description

Select a subset of events in a flowPlate. If a flowPlate and filter are supplied, then this function calls the Subset function from flowCore. Additionally, the plateCore version of Subset also makes it easy to filter individual flowFrames and keep the flowPlate structure.

Usage

```
## S4 method for signature 'flowPlate,ANY'
Subset(x, subset, select=NULL,...)
```

Arguments

`x`	A `flowPlate`
`subset`	A filter object
`select`	An optional vector of either sample names or Well.IDs.
`...`	optional arguments

Value

Returns a `flowPlate`

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

## Create a rectangle filter
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
```

```
## Apply the filter only to sample A01. The other flowFrames
## are not filtered.
fp <- Subset(fp,rectGate,'A01')
```

`applyControlGates` *Apply control gates to a flowPlate*

Description

Once setControlGates has been used to create gates for a flowPlate object, gates are applied to test samples using applyControlGates. The applyControlGates function is separated from setControlGates since gates may need be changed outside of setControlGates.

Usage

```
applyControlGates(data, gateType="Negative.Control", ...)
```

Arguments

`data`	A flowPlate dataset.
`gateType`	The type of gate to be applied to the flowPlate. Currently only "Negative.Control" gates are supported.
`...`	optional arguments

Value

Returns a `flowPlate` where the wellAnnotation now contains additional columns corresponding to total number of events in a well (Total.Count), the percentage of cells above background (Percent.Positive), and the number of positive cells (Positive.Count).

Author(s)

Errol Strain

See Also

See Also `setControlGates`

Examples

```
library(plateCore)
data(plateCore)

## Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

# Create a set of negative control gates and then apply them
fp <- setControlGates(fp,gateType="Negative.Control")
```

```
fp <- applyControlGates(fp,gateType="Negative.Control")

# Percent Positive and Counts columns are now in the wellAnnotation
head(wellAnnotation(fp))
```

`compensate` *Compensate a flowPlate to correct for the effects of spillover.*

Description

Flow samples are often stained with multiple types of fluorophores. Unfortunately, the emission spectra for these different fluorophores often overlap, and the signals must be corrected before proceeding with the analysis. Compensate adjusts for spillover using the method implemented in the package flowCore. Unlike flowCore, compensate only adjusts for the dyes/fluorophores listed in wellAnnotation.

Usage

```
## S4 method for signature 'flowPlate,ANY'
compensate(x, spillover)
```

Arguments

`x`	A flowPlate
`spillover`	The compensation matrix where the row and column names match the fluorescence channels of the flowPlate.

Value

Returns a compensated `flowPlate`.

Author(s)

Errol Strain

See Also

See Also `compensation-class`

Examples

```
library(plateCore)
data(plateCore)

# Create the compensation matrix
comp.mat <- spillover(x=compensationSet,unstained=sampleNames(compensationSet)[5],
patt=".*H",fsc="FSC-H",ssc="SSC-H",method="median")

## Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)
```

```
# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

# apply the compensation matrix
fp <- compensate(fp,comp.mat)
```

`compensationSet` *Sample Compensation Data Set*

Description

Sample Compensation Data Set

Usage

```
data(plateCore)
```

Format

The format is an object of class `flowSet` composed of 5 `flowFrames`. The `flowSet` consists of 4 stained and one unstained `flowFrames`. Peripheral Blood Mononucleocytes (PBMCs) were stained with FITC (Fluorescein isothiocyanate), PE (phycoerythrin), PerCp (Peridinin-chlorophyll), and APC (Allophycocyanin).

Author(s)

Errol Strain

Source

Sample data set from BD FACS CAP analysis.

See Also

See Also `compensation-class`

Examples

```
library(plateCore)
data(plateCore)

# Create the compensation matrix
comp.mat <- spillover(x=compensationSet,unstained=sampleNames(compensationSet)[5],
patt=".*H",fsc="FSC-H",ssc="SSC-H",method="median")
```

densityplot *One-dimensional density plots for flowPlates*

Description

This function is a modified version of densityplot from the flowViz package that allows for multiple flowFrames per panel. flowViz densityplot plots the density curves in a one per panel style, while the flowPlate densityplot can overlay densities.

Usage

```
## S4 method for signature 'formula,flowPlate'
densityplot(x, data, xlab,
    prepanel=prepanel.densityplot.flowPlate,
    panel = panel.densityplot.flowPlate,
    as.table=TRUE,
    filterResult=NULL,
    ...)
```

Arguments

x	A formula describing the layout of the plots.
data	A flowPlate.
xlab	Label for the x-axis
prepanel	Lattice-flowViz prepanel function.
panel	Lattice-flowViz panel function.
as.table	Defaults to table layout.
filterResult	filterResult can either take the character string "Negative.Control" and have the negative control wells added to the panels, or if filterResult is a flowFrame then the density curve for the flowFrame will be added to each panel.
...	optional arguments Other arguments are identical to densityPlot from flowViz.

See Also

flowViz::densityplot

Examples

```
# Load the plateCore package and data
library(plateCore)
data(plateCore)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

# Overlay the first 3 flowFrames.  If the groups argument was
# omitted, then the flowFrames would be combined into a single
# density curve.
densityplot(~ `FSC-H`, fp[1:3], groups=name, auto.key=TRUE)
```

`fixAutoFl` *Correct for the effects of cell size (FSC) on autofluorescence*

Description

The `fixAutoFl` function uses the method of Hahne et al. 2006 (Genome Biology) to fit a robust, log-log linear regression to the fluorescence channel of interest versus forward scatter (FSC). The current implementation scales the corrected data so the median fluorescence intensity (MFI) is the same before and after `fixAutoFl`.

Usage

```
fixAutoFl(fp,fsc="FSC.A",chanCols,unstain,...)
```

Arguments

`fp`	A `flowPlate`
`chanCols`	Selected channels to correct for autofluorescence.
`unstain`	Name(s) of the unstained samples. The function will try to find samples with Sample.Type="Unstained" if no names are given. If there are multiple unstained samples the function will average the slopes.
`fsc`	Name of the FSC parameter.
`...`	optional arguments

Value

Returns a flowPlate with autofluorescence due to cell size (FSC) corrected.

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

## Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)

## Create a flowPlate object from the platePBMC and the wellAnnotation
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

## Correct for autofluorescence in FL1.H-FL4.H
fp <- fixAutoFl(fp,fsc="FSC-H",chanCols=c("FL1-H","FL2-H","FL3-H","FL4-H"))
```

flowPlate-class *The flowPlate class.*

Description

flowPlates are the basic data containers for the plateCore package. A flowPlate is essentially a flowSet-class, plus a data.frame describing the layout of the plate and contents of individual wells.

Slots

plateName: A character string containing the name of the plate.

plateSet: A flowSet-class containing FCS event data. Prior to creating a flowPlate, the FCS files are first read into a flowSet-class using read.flowSet.

wellAnnotation: A data.frame describing the layout of the plate. Each row describes one channel for a well.

Methods

[, [[Subsetting. x[i] where i is either a scalar or character corresponding to a sample name, returns a flowPlate object, and x[[i]] a flowFrame object.

Usage:

```
flowSet[i]
flowSet[[i]]
```

Author(s)

Errol Strain, Florian Hahne, Perry Haaland

Examples

```
library(plateCore)
data(plateCore)

##Look at the wellAnnotation
wellAnnotation[1:4,]

## Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)

## Create a flowPlate object from the platePBMC and the wellAnnotation
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

## Subset the flowPlate, creating another flowPlate
fpSmall <- fp["A01"]

## Extract a flowFrame from a flowPlate
ff <-fp[["A01"]]
```

`flowPlate` *Create a flowPlate*

Description

Constructor for a `flowPlate` object. `sampleNames` for the `flowSet` should match the Well.Id column of `wellAnnotation`. Well.Ids must be unique to `sampleNames`, which is usually ensured by using the 3 character designations for wells (e.g. "A01","A02",...,"H12").

Usage

```
flowPlate(data,wellAnnotation,plateName,...)
```

Arguments

`data`	`flowSet` object to be made into a `flowPlate`
`wellAnnotation`	`data.frame` describing the layout and contents of the `flowPlate`
`plateName`	Name of the `flowPate`, should be unique within the set `flowPlates` under consideration
`...`	optional arguments

Value

Returns a `flowPate` object.

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

## Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)

## Create a flowPlate object from the platePBMC and the wellAnnotation
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

## Subset the flowPlate, creating another flowPlate
fpSmall <- fp["A01"]

## Extract a flowFrame from a flowPlate
ff <-fp[["A01"]]

## Retrieve sample names from flowPlate
sampNames <- sampleNames(fp)

## Retrieve the annotatedDataFrame describing the flowPlate
adf <- phenoData(fp)
```

fpbind *Merge multiple flowPlates into a single virtual flowPlate*

Description

A function to combine multiple `flowPlates` into a single `flowPlate` object. The `plateName` identifiers for the `flowPlates` must be unique within the set to be bound, otherwise the bind will fail.

Usage

```
fpbind(p1, p2, ...)
```

Arguments

`p1`	First `flowPlate`
`p2`	Second `flowPlate`
`...`	Additional `flowPlates`

Value

Returns a `flowPlate`

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

# Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)

# Create a flowPlate object from the platePBMC and the wellAnnotation
fp1 <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")
fp2 <- flowPlate(pbmcPlate,wellAnnotation,plateName="P2")

# Combine the plates.
virtPlate <- fpbind(fp1,fp2)
```

`getGroups` *Retrieve Negative control groups from a flowPlate*

Description

Retrieve a list of negative control-based groups from a flowPlate, based on the information in wellAnnotation.

Usage

```
getGroups(data,type="Negative.Control",chan, ...)
```

Arguments

`data`	A flowPlate dataset.
`type`	Currently only Negative.Control groups are supported.
`chan`	Fluorescence channel of interest ("FL1-H", "PE-H", etc.)
`...`	optional arguments

Value

Returns a list of groups, where each group contains a single negative control well and the associated test well for a particular channel.

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

# Create a set of negative control gates and then apply them
negCon <- getGroups(fp,chan="FL1-H")

negCon[1:2]
```

gutterPlot *gutterPlot*

Description

A Quality Control plot to check the number of events in each channel that are at either their minimum or maximum value. A large number of these events may indicate a problem with the sample.

Usage

```
gutterPlot(fp, chans = c("FSC-H", "SSC-H", "FL1-H", "FL2-H", "FL3-H", "FL4-H"),
```

Arguments

fp	A flowPlate.
chans	Channels of interest to show on the gutterPlot.
...	optional arguments

Value

Creates a plot where the x-axis is the different wells in a flowPlate, and the y-axis is the fraction of events at the boundary.

Author(s)

Jon Gosink and Errol Strain

Examples

```
library(plateCore)
data(plateCore)

## Create a flowPlate
fp <- flowPlate(pbmcPlate,wellAnnotation,"p1001")

gutterPlot(fp,chans=c("FSC-H","SSC-H","FL1-H","FL2-H","FL3-H","FL4-H"))
```

---

mfiPlot *mfiPlot*

Description

A Quality Control plot that shows the MFI Ratio versus the percentage of positive cells in a flowPlate. The robust logistic regression is performed using gmlrob from the robustbase package.

Usage

```
mfiPlot(fp, thresh=2, Sample.Type="Test",Events="Percentage", ...)
```

Arguments

fp	A flowPlate.
thresh	Points more than "thresh" number of standard deviations away from the best fit line will be colored red.
Sample.Type	Type of sample to show on plot. Defaults to "Test"
Events	The robust logistic regression can be performed using either the percentage of events above the negative control gate ("Percentage") or the actual number of events above the gate ("Actual").
...	optional arguments to plot and points.

Value

Creates a plot where the x-axis is MFI Ratio and the y-axis is the percentage of cells above the negative control gate.

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

## Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

# Create a set of negative control gates and then apply them
fp <- setControlGates(fp,gateType="Negative.Control")
fp <- applyControlGates(fp,gateType="Negative.Control")

# Compute summary statistics
fp <- summaryStats(fp)

## Create an MFI plot
mfiPlot(fp,thresh=2.5,xlab="MFI Ratio (Test MFI / Isotype MFI)",xlim=c(0.1,250),
ylab="Percentage of cells above the isotype gate",pch=23)
```

%on% *Methods for Function %on% in Package 'plateCore'*

Description

This operator is used to construct a transformFilter that first applies a transformList to the data before applying the filter operation to a flowPlate.

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

## Create a rectangle filter
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))

xyplot(`FL1-H` ~ `FSC-H` | as.factor(name),
transform("FL1-H"=log10) %on% fp, smooth=FALSE, filter=rectGate, displayFilter=FALSE)
```

---

panel.densityplot.flowPlate

Lattice-flowViz style panel function for flowPlate densityplot.

Description

This function should not be called directly, use densityplot.

Usage

```
panel.densityplot.flowPlate(x,
frames, channel, wellAnnotation,
groups=NULL,
subscripts,
col = superpose.symbol$col,
col.points = col,
col.line = col,
filterResult=NULL,
...)
```

Arguments

x	character
frames	flowFrames
channel	channel of interest
wellAnnotation	wellAnnotation data.frame
groups	density plot groups parameter
subscripts	densityplot subscripts parameter
col	densityplot col parameter
col.points	densityplot col.points parameter
col.line	densityplot col.line parameter
filterResult	densityplot filterResult parameter
...	optional arguments

Author(s)

Errol Strain

See Also

See Also densityplot

Examples

```
# Load the plateCore package and data
library(plateCore)
data(plateCore)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

# Overlay the first 3 flowFrames.  If the groups argument was
# omitted, then the flowFrames would be combined into a single
# density curve.
densityplot(~ `FSC-H`, fp[1:3], groups=name, auto.key=TRUE)
```

---

panel.xyplot.flowPlate

Lattice-flowViz style panel function for flowPlate xyplot.

Description

This function should not be called directly, use xyplot.

Usage

```
panel.xyplot.flowPlate(x,
frames,
channel.x, channel.y,
channel.x.name, channel.y.name,
filter = NULL,
filterResults = NULL,
displayFilter = TRUE,
pch, smooth,
wellAnnotation = NULL,
col = superpose.symbol$col,
...)
```

Arguments

x	character
frames	flowFrames
channel.x	xyplot channel.x parameter
channel.y	xyplot channel.y parameter

`channel.x.name`
: xyplot channel.x.name parameter

`channel.y.name`
: xyplot channel.y.name parameter

`filter`
: xyplot filter parameter

`filterResults`
: xyplot filterResults parameter

`displayFilter`
: xyplot displayFilter parameter

`pch`
: xyplot pch parameter

`smooth`
: xyplot smooth parameter

`wellAnnotation`
: wellAnnotation data.frame

`col`
: xyplot col parameter

`...`
: optional arguments

Author(s)

Errol Strain

See Also

See Also `xyplot`

Examples

```
library(plateCore)
data(plateCore)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

## Create a rectangle filter
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))

xyplot(`SSC-H` ~ `FSC-H` | as.factor(name),
fp[1], smooth=FALSE, filter=rectGate, displayFilter=FALSE)
```

`pbmcPlate` *pbmcPlate Data Set*

Description

One 96-well plate from a BD FACS CAP analysis of Peripheral Blood Mononucleocyte (PMBC) cells.

Usage

```
data(plateCore)
```

Format

The format is an object of class `flowSet` composed of 96 `flowFrames`. Each `flowFrame` corresponds to one well from the plate.

Details

BD FACS CAP (http://www.bd.com/technologies/discovery_platform/BD_FACS_CAP.asp) is a platform for screening a large number of antibodies (200+) on human samples. Antibodies are arrayed 3-per well on a 96-well plate, along with the appropriate controls. In this experiment, an early version of FACS CAP was used to screen PBMCs from 2 donors for 189 different human cell surface markers. The complete data set is available from the FICCS site shown below. The `pbmcPlate` include with plateCore is a lymphocyte enriched subset of one of the replicate plates for donor 1.

Source

Complete dataset available at http://www.ficcs.org/software.html#Data_Files, the Flow Informatics and Computational Cytometry Society website (FICCS)

References

Add reference for plateCore paper (when/if published).

---

plateCore-package *plateCore: A Bioconductor package for high throughput analysis of flow cytometry data*

Description

`plateCore` is a Bioconductor packaged created to make processing and analysis of large, complex flow datasets in R easier. High throughput flow studies are often run in a 96 or 384-well plate format, with a number of different samples, controls, and antibodies-dye conjugates present on the plate. Analyzing the output from the cytometer requires keeping track of the contents of each well, matching sample wells with control wells, gating each well/channel separately, making the appropriate plots, and summarizing the results. `plateCore` extends the `flowCore` and `flowViz` packages to work on `flowPlate` objects that represent these large flow datasets. For those familiar with `flowCore` and `flowViz`, the gating (filtering), transformation, and other data manipulations for `flowPlates` are very similar to `flowSets`.

Details

Package:	plateCore
Type:	Package
Version:	1.2.1
Date:	2009-06-29

Author(s)

Errol Strain, Florian Hahne, and Perry Haaland Maintainer: Errol Strain <estrain@gmail.com>

References

Insert flowCore and flowViz publications.

Examples

```
library(plateCore)
data(plateCore)

## Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)

## Create a flowPlate object from the platePBMC and the wellAnnotation
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")
```

---

plateSet *Get the flowSet from a flowPlate object*

Description

A function to retrieve the `flowSet` from a `flowPlate`.

Usage

```
plateSet(fp, ...)
```

Arguments

`fp`	A `flowPlate`
`...`	optional arguments

Value

Returns a `flowSet`

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

## Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)

## Create a flowPlate object from the platePBMC and the wellAnnotation
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

## Retrieves the flowSet
fs <- plateSet(fp)
```

`plotPlate` *plotPlate*

Description

Make a row vs. column plot of a plate, where the wells are colored according to some value of choice (number of events, median signal intensity, percent positive, etc.).)

Usage

```
plotPlate(fp, x = NA, method = "median", main, col, values, width = 1, na.action
```

Arguments

`fp`	A `flowPlate`.
`x`	A `character` indicating the variable of interest. Valid choices are "events", any single channel name (e.g. FSC-H, SSC-H, FL1-H, etc.), or vector of channel names if the method is mahalanobis.
`method`	Valid choices are mean, median, sd, mad, mahalanobis, or one of the numeric columns in the wellAnnotation data.frame (e.g. Percent.Positive, Positive.Count, MFI, MFI.Ratio)
`main`	Main text for the plot
`col`	Character vector of two colors.
`values`	Optional list of values, with names corresponding to sampleNames, that will be used for plotting.
`width`	Width of the well.
`na.action`	Handling of NA values, either "zero" or "omit".
`...`	optional arguments

Value

Plots the plate to the standard output.

Author(s)

Jon Gosink and Errol Strain

References

The original version of this plot came from the `prada` package.

Examples

```
library(plateCore)
data(plateCore)

## Create a flowPlate
fp <- flowPlate(pbmcPlate,wellAnnotation,"p1001")

plotPlate(transform("FL1-H"=log10) %on% fp,x="FL1-H",method="mean",col=c("yellow", "darkb
```

---

prepanel.densityplot.flowPlate

Lattice-flowViz style panel function for flowPlate densityplot.

Description

This function should not be called directly, use `densityplot`.

Usage

```
prepanel.densityplot.flowPlate(x, frames, channel,...)
```

Arguments

`x`	Character
`frames`	flowFrames
`channel`	Character string for channel name.
`...`	optional arguments

Author(s)

Errol Strain

See Also

See Also `densityplot`

Examples

```
# Load the plateCore package and data
library(plateCore)
data(plateCore)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

# Overlay the first 3 flowFrames.  If the groups argument was
# omitted, then the flowFrames would be combined into a single
# density curve.
densityplot(~ `FSC-H`, fp[1:3], groups=name, auto.key=TRUE)
```

`prepanel.xyplot.flowPlate` *Lattice-flowViz style panel function for flowPlate xyplot.*

Description

This function should not be called directly, use `xyplot`.

Usage

```
prepanel.xyplot.flowPlate(x, frames, channel.x, channel.y,...)
```

Arguments

`x`	Character
`frames`	flowFrames
`channel.x`	Character string for channel name.
`channel.y`	Character string for channel name.
`...`	optional arguments

Author(s)

Errol Strain

See Also

See Also `xyplot`

Examples

```
library(plateCore)
data(plateCore)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

## Create a rectangle filter
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))

xyplot(`SSC-H` ~ `FSC-H` | as.factor(name),
fp[1], smooth=FALSE, filter=rectGate, displayFilter=FALSE)
```

`setControlGates` *Create control gates for a flowPlate*

Description

A function to estimate the threshold between positive and negative cells. This threshold corresponds to a one-dimensional gate, and cells above the gate are considered positive. The default value of `numMads=5` generally works well on the linear scale, but will need to be adjusted for transformed data. If each well contains a large number of events for the cell type of interest (>1000), then using the 99.5th quantile usually gives similar values.

Usage

```
setControlGates(data, gateType, threshType="MAD", numMads=5, isoquantile=.995, .
```

Arguments

`data`	A `flowPlate`
`gateType`	The type of gate to be set. Currently only "Negative.Control" gates are supported.
`threshType`	Values can be either "MAD", for median absolute deviation based gating, or "isoQuant" for quantile based gating.
`numMads`	Number of median absolute deviations above the median to set the initial gate.
`isoquantile`	Quantile setting for "isoQuant" threshType.
`...`	optional arguments.

Value

Returns a `flowPlate`

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

## Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

# Create a set of negative control gates and then apply them
fp <- setControlGates(fp,gateType="Negative.Control")

# There should now be a Negative.Control.Gate column in wellAnnotation
head(wellAnnotation(fp))
```

summaryStats *Compute summary statistics on a flowPlate*

Description

This function computes the median fluorescence intensity (MFI) and the MFI ratio (ratio of test well MFI to negative control MFI) for each well/channel in a `flowPlate`. The predicted percent positive (Predict.PP) and gate score (Gate.Score) come from a robust logistic regression of the MFI ratio to either the percentage of positive cells or the actual count of positive cells. Predict.PP is the estimated percent positive based on the MFI ratio and Gate.Score is the number of standardized residuals the sample data point is away from the best fit line. The glmrob function from the robustbase package is used for the regression. Results from summaryStats are stored in the `wellAnnotation data.fame`.

Usage

```
summaryStats(data, Events="Percentage", ...)
```

Arguments

`data`	A `flowPlate`
`Events`	The robust logistic regression can be performed using either the percentage of events above the negative control gate ("Percentage") or the actual number of events above the gate ("Actual").
`...`	optional arguments

Value

Returns a `flowPlate`

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

## Get the lymphocytes
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))
pbmcPlate <- Subset(pbmcPlate, rectGate)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

# Create a set of negative control gates and then apply them
fp <- setControlGates(fp,gateType="Negative.Control")
fp <- applyControlGates(fp,gateType="Negative.Control")

# Compute summary statistics
fp <- summaryStats(fp)
```

```
# There should now be MFI and MFI.ratio columns in the wellAnnotation
head(wellAnnotation(fp))
```

`wellAnnotation` *Retrieve a data.frame describing the content of a flowPlate*

Description

wellAnnotation returns the tall `data.frame` describing the layout of a `flowPate`, where each row corresponds to one well-channel.

Usage

```
wellAnnotation(fp, ...)
```

Arguments

`fp`	A flowPlate dataset.
`...`	optional arguments

Value

Returns a `data.frame`.

Author(s)

Errol Strain

Examples

```
library(plateCore)
data(plateCore)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

# Look at the top of wellAnnotation
head(wellAnnotation(fp))
```

`xyplot` *Scatter plots (dotplots) for flowPlates.*

Description

A function to create dotplots, and smoothed scatter plots, from flowPlates. This function is a slightly modified version of xyplot from flowViz. The flowPlate xyplot allows users to overlay plots of test samples versus controls, and makes creating informative flowStrips easier. Refer to the documentation for xyplot from flowViz and lattice for more detailed information.

Usage

```
## S4 method for signature 'formula,flowPlate'
xyplot(x, data, xlab, ylab,
as.table = TRUE,
prepanel = prepanel.xyplot.flowPlate,
panel = panel.xyplot.flowPlate,
pch = ".", smooth = TRUE,
filter = NULL,
filterResults = NULL,
displayFilter = TRUE,
flowStrip=NULL,
flowStripCex=1,
strip=function(...,style=1) strip.default(...,style=1),
...)
```

Arguments

`x`	A formula describing the layout of the plots. Plots for flowPlates usually condition on either `as.factor(name)` or `as.factor(Well.Id)` since only one flowFrame can be shown on each panel (with the exception of Negative.Control overlays).
`data`	A flowPlate.
`xlab`	Label for x-axis.
`ylab`	Label for y-axis.
`as.table`	Defaults to table layout.
`prepanel`	Lattice-flowViz prepanel function.
`panel`	Lattice-flowViz panel function.
`pch`	Plotting character.
`smooth`	Plot a smoothed scatterplot by default.
`filter`	A flowCore filter to apply to each flowFrame.
`filterResults`	If filterResults="Negative.Control", the negative control wells corresponding to a test well are overlayed in the test well plots.
`displayFilter`	Defaults to displaying filter on the plot.

flowStrip	Character vector indicating additional information to be printed on the strip. Values can include any combination of "Well.Id","MFI","MFI.Ratio", and "Percent.Positive".
flowStripCex	Font size for the flowStrip.
strip	Lattice strip function.
...	Optional arguments

Author(s)

Errol Strain

See Also

flowViz::xyplot

Examples

```
library(plateCore)
data(plateCore)

# Create a flowPlate from the sample data in plateCore
fp <- flowPlate(pbmcPlate,wellAnnotation,plateName="P1")

## Create a rectangle filter
rectGate <- rectangleGate("FSC-H"=c(300,700),"SSC-H"=c(50,400))

xyplot(`SSC-H` ~ `FSC-H` | as.factor(name),
fp[1], smooth=FALSE, filter=rectGate, displayFilter=FALSE)
```

Index