Package 'flowClust'

April 11, 2018

Type Package

Title Clustering for Flow Cytometry

Version 3.16.0

Author Raphael Gottardo <raph@stat.ubc.ca>, Kenneth Lo <c.lo@stat.ubc.ca>, Greg
Finak <gfinak@fhcrc.org>

Maintainer Greg Finak <gfinak@fhcrc.org>, Mike Jiang <wjiang2@fhcrc.org>

Depends R(>= 2.5.0)

Imports BiocGenerics, MCMCpack, methods, Biobase, graph, RBGL,
ellipse, flowViz, flowCore, clue, mnormt

Suggests testthat, flowWorkspace, flowWorkspaceData

Enhances

Description Robust model-based clustering using a t-mixture model with
Box-Cox transformation. Note: users should have GSL installed.
Windows users: 'consult the README file available in the inst
directory of the source distribution for necessary
configuration instructions'.

Collate 'SetClasses.R' 'SetMethods.R' 'SimulateMixture.R'
'flowClust.R' 'coerce.R' 'flowClust-package.R' 'hist.R'
'miscellaneous.R' 'mkPriors.R' 'peakMatch.R' 'plot.R' 'split.R'

License Artistic-2.0

biocViews Clustering, Visualization, FlowCytometry

SystemRequirements GNU make

RoxygenNote 5.0.1

NeedsCompilation yes

R topics documented:

---

flowClust-package   *Clustering for Flow Cytometry*

Description

Robust model-based clustering using a t mixture model with Box-Cox transformation.

Details

Package:	flowClust
Type:	Package
Version:	2.0.0
Depends:	R(>= 2.5.0), methods, mnormt, mclust, ellipse, Biobase, flowCore
Collate:	SetClasses.R SetMethods.R plot.R flowClust.R SimulateMixture.R
biocViews:	Clustering, Statistics, Visualization
License:	Artistic-2.0
Built:	R 2.6.1; universal-apple-darwin8.10.1; 2008-03-26 20:54:42; unix

Index

list(list("box")) Box-Cox Transformation

list(list("density,flowClust-method")) Grid of Density Values for the Fitted t Mixture Model with Box-Cox Transformation

list(list("dmvt")) Density of the Multivariate t Distribution with Box-Cox Tranformation

list(list("dmvtmix")) Density of the Multivariate t Mixture Distribution with Box-Cox Tranformation

box 3

list(list("flowClust")) Robust Model-based Clustering for Flow Cytometry

list(list("hist.flowClust")) 1-D Density Plot (Histogram) of Clustering Results

list(list("Map,flowClust-method")) Cluster Assignment Based on Clustering Results

list(list("miscellaneous")) Various Functions for Retrieving Information from Clustering Results

list(list("plot,flowClust-method")) Scatterplot of Clustering Results

list(list("plot,flowDens-method")) Contour or Image Plot of Clustering Results

list(list("plot,flowFrame,tmixFilterResult-method")) Scatterplot / 1-D Density Plot of Filtering (Clustering) Results

list(list("rbox")) Reverse Box-Cox Transformation

list(list("ruleOutliers,flowClust-method")) Showing or Modifying the Rule used to Identify Outliers

list(list("show,flowClust-method")) Show Method for `flowClust` / `tmixFilterResult` Object

list(list("show,tmixFilter-method")) Show Method for `tmixFilter` Object

list(list("SimulateMixture")) Random Generation from a t Mixture Model with Box-Cox Transformation

list(list("split,flowClust-method")) Splitting Data Based on Clustering Results

list(list("Subset,flowClust-method")) Subsetting Data Based on Clustering Results

list(list("summary,flowClust-method")) Summary Method for `flowClust` Object

list(list("tmixFilter")) Creating Filters and Filtering Flow Cytometry Data

Note

Further information is available in the vignette.

Author(s)

Raphael Gottardo <raph@stat.ubc.ca>, Kenneth Lo <c.lo@stat.ubc.ca>

Maintainer: Raphael Gottardo <raph@stat.ubc.ca>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

---

box   *Box-Cox Transformation*

Description

This function performs Box-Cox transformation on the inputted data matrix.

Usage

```
box(data, lambda)
```

Arguments

data   A numeric vector, matrix or data frame of observations. Negative data values
                are permitted.

lambda   The transformation to be applied to the data. If negative data values are present,
                lambda has to be positive.

Details

To allow for negative data values, a slightly modified version of the original Box-Cox (1964) is used here. This modified version originated from Bickel and Doksum (1981), taking the following form:

$$f(y) = \frac{\operatorname{sgn}(y)|y|^{\lambda} - 1}{\lambda}$$

When negative data values are involved, the transformation parameter, λ, has to be positive in order to avoid discontinuity across zero.

Value

A numeric vector, matrix or data frame of the same dimension as data is returned.

References

Bickel, P. J. and Doksum, K. A. (1981) An Analysis of Transformations Revisited. *J. Amer. Statist. Assoc.* **76**(374), 296-311.

Box, G. E. P. and Cox, D. R. (1964) An Analysis of Transformations. *J. R. Statist. Soc. B* **26**, 211-252.

See Also

[rbox](#)

Examples

```
data(rituximab)
library(flowCore)
data <- exprs(rituximab)
summary(data)
# Transform data using Box-Cox with lambda=0.3
dataTrans <- box(data, 0.3)
# Reverse transform data; this should return back to the original rituximab data
summary(rbox(dataTrans, 0.3))
```

---

criterion   *Various Functions for Retrieving Information from Clustering Results*

Description

Various functions are available to retrieve the information criteria (criterion), the posterior probabilities of clustering memberships z (posterior), the "weights" u (importance), the uncertainty (uncertainty), and the estimates of the cluster proportions, means and variances (getEstimates) resulted from the clustering (filtering) operation.

Usage

```
criterion(object, ...)

## S4 method for signature 'flowClust'
criterion(object, type = "BIC")

## S4 method for signature 'flowClustList'
criterion(object, type = "BIC", max = FALSE,
  show.K = FALSE)

criterion(object) <- value

## S4 replacement method for signature 'flowClustList,character'
criterion(object) <- value

posterior(object, assign = FALSE)

importance(object, assign = FALSE)

uncertainty(object)

getEstimates(object, data)
```

Arguments

object	Object returned from [flowClust](#) or [filter](#). For the replacement method of criterion, the object must be of class [flowClustList](#) or [tmixFilterResultList](#).
type, value	A character string stating the criterion used to choose the best model. May take either "BIC" or "ICL".
max	whether criterion should return the max value
show.K	whether criterion should return K
assign	A logical value. If TRUE, only the quantity (z for posterior or u for importance) associated with the cluster to which an observation is assigned will be returned. Default is FALSE, meaning that the quantities associated with all the clusters will be returned.
data	A numeric vector, matrix, data frame of observations, or object of class flowFrame; an optional argument. This is the object on which [flowClust](#) or [filter](#) was performed.
...	Further arguments. Currently this is type, a character string. May take "BIC", "ICL" or "logLike", to specify the criterion desired.

Details

These functions are written to retrieve various slots contained in the object returned from the clustering operation. criterion is to retrieve object@BIC, object@ICL or object@logLike. It replacement method modifies object@index and object@criterion to select the best model according to the desired criterion. posterior and importance provide a means to conveniently retrieve information stored in object@z and object@u respectively. uncertainty is to retrieve object@uncertainty. getEstimates is to retrieve information stored in object@mu (transformed back to the original scale) and object@w; when the data object is provided, an approximate variance

estimate (on the original scale, obtained by performing one M-step of the EM algorithm without taking the Box-Cox transformation) will also be computed.

Value

Denote by K the number of clusters, N the number of observations, and P the number of variables. For posterior and importance, a matrix of size $N \times K$ is returned if assign=FALSE (default). Otherwise, a vector of size N is outputted. uncertainty always outputs a vector of size N. getEstimates returns a list with named elements, proportions, locations and, if the data object is provided, dispersion. proportions is a vector of size P and contains the estimates of the K cluster proportions. locations is a matrix of size $K \times P$ and contains the estimates of the K mean vectors transformed back to the original scale (i.e., rbox(object@mu,object@lambda)). dispersion is an array of dimensions $K \times P \times P$, containing the approximate estimates of the K covariance matrices on the original scale.

Note

When object@nu=Inf, the Mahalanobis distances instead of the "weights" are stored in object@u. Hence, importance will retrieve information corresponding to the Mahalanobis distances. the assign argument is set to TRUE, only the quantities corresponding to assigned observations will be returned. Quantities corresponding to unassigned observations (outliers and filtered observations) will be reported as NA. Hence, A change in the rule to call outliers will incur a change in the number of NA values returned.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

flowClust, filter, Map

---

dmvt   *Density of the Multivariate t Distribution with Box-Cox Tranformation*

Description

This function computes the densities at the inputted points of the multivariate t distribution with Box-Cox transformation.

Usage

```
dmvt(x, mu, sigma, nu, lambda, log = FALSE)
```

Arguments

x	A matrix or data frame of size $N \times P$, where N is the number of observations and P is the dimension. Each row corresponds to one observation.
mu	A numeric vector of length P specifying the mean.
sigma	A matrix of size $P \times P$ specifying the covariance matrix.
nu	The degrees of freedom used for the t distribution. If nu=Inf, Gaussian distribution will be used.
lambda	The Box-Cox transformation parameter. If missing, the conventional t distribution without transformation will be used.
log	A logical value. If TRUE then the logarithm of the densities is returned.

Value

A list with the following components:

value	A vector of length N containing the density values.
md	A vector of length N containing the Mahalanobis distances.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

---

dmvtmix	*Density of the Multivariate t Mixture Distribution with Box-Cox Tranformation*

Description

This function computes the densities at the inputted points of the multivariate t mixture distribution with Box-Cox transformation.

Usage

```
dmvtmix(x, w, mu, sigma, nu, lambda, object, subset, include, log = FALSE)
```

Arguments

x	A matrix or data frame of size $N \times P$, where N is the number of observations and P is the dimension. Each row corresponds to one observation.
w	A numeric vector of length K containing the cluster proportions.
mu	A matrix of size $K \times P$ containing the K mean vectors.
sigma	An array of size $K \times P \times P$ containing the K covariance matrices.
nu	A numeric vector of length K containing the degrees of freedom used for the t distribution. If only one value is specified for nu, then it is used for all K clusters. If nu=Inf, Gaussian distribution will be used.
lambda	The Box-Cox transformation parameter. If missing, the conventional t distribution without transformation will be used.

object	An optional argument. If provided, it's an object returned from [flowClust](#), and the previous arguments will be assigned values from the corresponding slots of `object`.
subset	An optional argument. If provided, it's a numeric vector indicating which variables are selected for computing the densities. If `object` is provided and `object@varNames` is not `NULL`, then a character vector containing the names of the variables is allowed.
include	An optional argument. If provided, it's a numeric vector specifying which clusters are included for computing the densities.
log	A logical value. If `TRUE` then the logarithm of the densities is returned.

Value

A vector of length N containing the density values.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

---

flowClust-class   *Robust Model-based Clustering for Flow Cytometry*

Description

This function performs automated clustering for identifying cell populations in flow cytometry data. The approach is based on the tmixture model with the Box-Cox transformation, which provides a unified framework to handle outlier identification and data transformation simultaneously.

Usage

```
flowClust(x, expName = "Flow Experiment", varNames = NULL, K, nu = 4,
  lambda = 1, trans = 1, min.count = 10, max.count = 10, min = NULL,
  max = NULL, randomStart = 0, prior = NULL, usePrior = "no",
  criterion = "BIC", ...)
```

Arguments

x	A numeric vector, matrix, data frame of observations, or object of class `flowFrame`. Rows correspond to observations and columns correspond to variables.
expName	A character string giving the name of the experiment.
varNames	A character vector specifying the variables (columns) to be included in clustering. When it is left unspecified, all the variables will be used.
K	An integer vector indicating the numbers of clusters.
nu	The degrees of freedom used for the t distribution. Default is 4. If `nu=Inf`, Gaussian distribution will be used.
lambda	The initial transformation to be applied to the data.
trans	A numeric indicating whether the Box-Cox transformation parameter is estimated from the data. May take 0 (no estimation), 1 (estimation, default) or 2 (cluster-specific estimation).

min.count	An integer specifying the threshold count for filtering data points from below. The default is 10, meaning that if 10 or more data points are smaller than or equal to min, they will be excluded from the analysis. If min is NULL, then the minimum of data as per each variable will be used. To suppress filtering, set it as -1.
max.count	An integer specifying the threshold count for filtering data points from above. Interpretation is similar to that of min.count.
min	The lower boundary set for data filtering. Note that it is a vector of length equal to the number of variables (columns), implying that a different value can be set as per each variable.
max	The upper boundary set for data filtering. Interpretation is similar to that of min.
randomStart	A numeric value indicating how many times a random parition of the data is generated for initialization. The default is 0, meaning that a deterministic partition based on kmeans clustering is used. A value of 10 means random partitions of the data will be generated, each of which is followed by a short EM run. The partition leading to the highest likelihood value will be adopted to be the initial partition for the eventual long EM run.
prior	The specification of the prior. Used if usePrior="yes"
usePrior	Argument specifying whether or not the prior will be used. Can be "yes","no","vague". A vague prior will be automatically specified if usePrior="vague"
criterion	A character string stating the criterion used to choose the best model. May take either ″BIC″ or ″ICL″. This argument is only relevant when length(K)>1. Default is "BIC".
...	other arguments: B: The maximum number of EM iterations.Default is 500.
	tol: The tolerance used to assess the convergence of the EM. default is 1e-5.
	nu.est: A numeric indicating whether nu is to be estimated or not. May take 0 (no estimation, default), 1 (estimation) or 2 (cluster-specific estimation). Default is 0.
	level: A numeric value between 0 and 1 specifying the threshold quantile level used to call a point an outlier. The default is 0.9, meaning that any point outside the 90% quantile region will be called an outlier.
	u.cutoff: Another criterion used to identify outliers. If this is NULL, which is default, then level will be used. Otherwise, this specifies the threshold (e.g., 0.5) for u, a quantity used to measure the degree of "outlyingness" based on the Mahalanobis distance. Please refer to Lo et al. (2008) for more details.
	z.cutoff: A numeric value between 0 and 1 underlying a criterion which may be used together with level/u.cutoff to identify outliers. A point with the probability of assignment z (i.e., the posterior probability that a data point belongs to the cluster assigned) smaller than z.cutoff will be called an outlier. The default is 0, meaning that assignment will be made no matter how small the associated probability is, and outliers will be identified solely based on the rule set by level or cutoff.
	B.init: The maximum number of EM iterations following each random partition in random initialization. Default is the same as B.
	tol.init: The tolerance used as the stopping criterion for the short EM runs in random initialization. Default is 1e-2.
	seed: An integer giving the seed number used when randomStart>0.Default is 1.
	control: An argument reserved for internal use.

Details

Estimation of the unknown parameters (including the Box-Cox parameter) is done via an Expectation-Maximization (EM) algorithm. At each EM iteration, Brent's algorithm is used to find the optimal value of the Box-Cox transformation parameter. Conditional on the transformation parameter, all other estimates can be obtained in closed form. Please refer to Lo et al. (2008) for more details.

The **flowClust** package makes extensive use of the GSL as well as BLAS. If an optimized BLAS library is provided when compiling the package, the **flowClust** package will be able to run multi-threaded processes.

Various operations have been defined for the object returned from `flowClust`.

In addition, to facilitate the integration with the **flowCore** package for processing flow cytometry data, the `flowClust` operation can be done through a method pair (`tmixFilter` and `filter`) such that various methods defined in **flowCore** can be applied on the object created from the filtering operation.

Value

If K is of length 1, the function returns an object of class `flowClust` containing the following slots, where K is the number of clusters, N is the number of observations and P is the number of variables:

expName	Content of the `expName` argument.
varNames	Content of the `varNames` argument if provided; generated if available otherwise.
K	An integer showing the number of clusters.
w	A vector of length K, containing the estimates of the K cluster proportions.
mu	A matrix of size $K \times P$, containing the estimates of the K mean vectors.
sigma	An array of dimension $K \times P \times P$, containing the estimates of the K covariance matrices.
lambda	The Box-Cox transformation parameter estimate.
nu	The degrees of freedom for the t distribution.
z	A matrix of size $N \times K$, containing the posterior probabilities of cluster memberships. The probabilities in each row sum up to one.
u	A matrix of size $N \times K$, containing the "weights" (the contribution for computing cluster mean and covariance matrix) of each data point in each cluster. Since this quantity decreases monotonically with the Mahalanobis distance, it can also be interpreted as the level of "outlyingness" of a data point. Note that, when nu=Inf, this slot is used to store the Mahalanobis distances instead.
label	A vector of size N, showing the cluster membership according to the initial partition (i.e., hierarchical clustering if `randomStart=0` or random partitioning if `randomStart>0`). Filtered observations will be labelled as NA. Unassigned observations (which may occur since only 1500 observations at maximum are taken for hierarchical clustering) will be labelled as 0.
uncertainty	A vector of size N, containing the uncertainty about the cluster assignment. Uncertainty is defined as 1 minus the posterior probability that a data point belongs to the cluster to which it is assigned.
ruleOutliers	A numeric vector of size 3, storing the rule used to call outliers. The first element is 0 if the criterion is set by the `level` argument, or 1 if it is set by `u.cutoff`. The second element copies the content of either the `level` or `u.cutoff` argument. The third element copies the content of the `z.cutoff` argument. For instance, if

	points are called outliers when they lie outside the 90% quantile region or have assignment probabilities less than 0.5, then ruleOutliers is c(0, 0.9, 0.5). If points are called outliers only if their "weights" in the assigned clusters are less than 0.5 regardless of the assignment probabilities, then ruleOutliers becomes c(1, 0.5, 0).
flagOutliers	A logical vector of size N, showing whether each data point is called an outlier or not based on the rule defined by level/u.cutoff and z.cutoff.
rm.min	Number of points filtered from below.
rm.max	Number of points filtered from above.
logLike	The log-likelihood of the fitted mixture model.
BIC	The Bayesian Information Criterion for the fitted mixture model.
ICL	The Integrated Completed Likelihood for the fitted mixture model.

If K has a length >1, the function returns an object of class flowClustList. Its data part is a list with the same length as K, each element of which is a flowClust object corresponding to a specific number of clusters. In addition, the resultant flowClustList object contains the following slots:

index An integer giving the index of the list element corresponding to the best model as selected by criterion.
criterion The criterion used to choose the best model – either "BIC" or "ICL".

Note that when a flowClustList object is used in place of a flowClust object, in most cases the list element corresponding to the best model will be extracted and passed to the method/function call.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

summary, plot, density, hist, Subset, split, ruleOutliers, [Map](Map), SimulateMixture

Examples

```
library(flowCore)
data(rituximab)

### cluster the data using FSC.H and SSC.H
res1 <- flowClust(rituximab, varNames=c("FSC.H", "SSC.H"), K=1)

### remove outliers before proceeding to the second stage
# %in% operator returns a logical vector indicating whether each
# of the observations lies within the cluster boundary or not
rituximab2 <- rituximab[rituximab %in% res1,]
# a shorthand for the above line
```

```
rituximab2 <- rituximab[res1,]
# this can also be done using the Subset method
rituximab2 <- Subset(rituximab, res1)

### cluster the data using FL1.H and FL3.H (with 3 clusters)
res2 <- flowClust(rituximab2, varNames=c("FL1.H", "FL3.H"), K=3)
show(res2)
summary(res2)

# to demonstrate the use of the split method
split(rituximab2, res2)
split(rituximab2, res2, population=list(sc1=c(1,2), sc2=3))

# to show the cluster assignment of observations
table(Map(res2))

# to show the cluster centres (i.e., the mean parameter estimates
# transformed back to the original scale)
getEstimates(res2)$locations

### demonstrate the use of various plotting methods
# a scatterplot
plot(res2, data=rituximab2, level=0.8)
plot(res2, data=rituximab2, level=0.8, include=c(1,2), grayscale=TRUE,
    pch.outliers=2)
# a contour / image plot
res2.den <- density(res2, data=rituximab2)
plot(res2.den)
plot(res2.den, scale="sqrt", drawlabels=FALSE)
plot(res2.den, type="image", nlevels=100)
plot(density(res2, include=c(1,2), from=c(0,0), to=c(400,600)))
# a histogram (1-D density) plot
hist(res2, data=rituximab2, subset="FL1.H")

### to demonstrate the use of the ruleOutliers method
summary(res2)
# change the rule to call outliers
ruleOutliers(res2) <- list(level=0.95)
# augmented cluster boundaries lead to fewer outliers
summary(res2)

# the following line illustrates how to select a subset of data
# to perform cluster analysis through the min and max arguments;
# also note the use of level to specify a rule to call outliers
# other than the default
flowClust(rituximab2, varNames=c("FL1.H", "FL3.H"), K=3, B=100,
    min=c(0,0), max=c(400,800), level=0.95, z.cutoff=0.5)
```

---

flowClust.den   *generate the curve that reflects the tmixture fitting outcome*

Description

generate the curve that reflects the tmixture fitting outcome

Usage

```
flowClust.den(x, obj, subset, include)
```

Arguments

x	the numeric vector represents the x coordinates in plot
obj	the flowClust object
subset	An integer indicating which variable is selected for the plot. Alternatively, a character string containing the name of the variable is allowed if x@varNames is not NULL.
include	A numeric vector specifying which clusters are shown on the plot. By default, all clusters are included.

---

flowClust2Prior	*Generate a prior specification based on a flowClust model This function generates a prior specification based on a flowClust fit object It can be passed to a second round of flowClust() with usePrior="yes" The prior could be estimated from a single sample, for example, and then used to speed up the convergence for other samples.*

Description

Generate a prior specification based on a flowClust model This function generates a prior specification based on a flowClust fit object It can be passed to a second round of flowClust() with usePrior="yes" The prior could be estimated from a single sample, for example, and then used to speed up the convergence for other samples.

Usage

```
flowClust2Prior(x, kappa, Nt = NULL, addCluster = NULL)
```

Arguments

x	a flowClust fit object
kappa	is the fraction of equivalent observations by which to weight this prior relative to the flowClust model.
Nt	the number of total equivalent observation
addCluster	not currently supported

flowDens-class   *Grid of Density Values for the Fitted t Mixture Model with Box-Cox Transformation*

Description

This method constructs the flowDens object which is used to generate a contour or image plot.

Usage

```
## S4 method for signature 'flowClust'
density(x, data = NULL, subset = c(1, 2),
  include = 1:(x@K), npoints = c(100, 100), from = NULL, to = NULL)

## S4 method for signature 'flowClustList'
density(x, data = NULL, subset = c(1, 2),
  include = 1:(x@K), npoints = c(100, 100), from = NULL, to = NULL)
```

Arguments

x	Object returned from [flowClust](#) or from running filter on a flowFrame object.
data	A matrix, data frame of observations, or object of class flowFrame. This is the object on which flowClust or filter was performed. If this argument is not specified, the grid square upon which densities will be computed must be provided (through arguments from and to).
subset	A numeric vector of length two indicating which two variables are selected for the scatterplot. Alternatively, a character vector containing the names of the two variables is allowed if x@varNames is not NULL.
include	A numeric vector specifying which clusters are included to compute the density values. By default, all clusters are included.
npoints	A numeric vector of size two specifying the number of grid points in x (horizontal) and y (vertical) directions respectively.
from	A numeric vector of size two specifying the coordinates of the lower left point of the grid square. Note that, if this (and to) is not specified, data must be provided such that the range in the two variables (dimensions) selected will be used to define the grid square.
to	A numeric vector of size two specifying the co-ordinates of the upper right point of the grid square.

Details

The flowDens object returned is to be passed to the plot method for generating a contour or image plot.

Value

An object of class flowDens containing the following slots is constructed:

dx	A numeric vector of length npoints[1]; the x-coordinates of the grid points.
dy	A numeric vector of length npoints[2]; the y-coordinates of the grid points.
value	A matrix of size npoints[1] $\times$ npoints[2]; the density values at the grid points.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

See Also

[plot](#), [flowClust](#)

---

hist.flowClust	*1-D Density Plot (Histogram) of Clustering Results*

Description

This method generates a one-dimensional density plot for the specified dimension (variable) based on the robust model-based clustering results. A histogram of the actual data or cluster assignment is optional for display.

Usage

```
## S3 method for class 'flowClust'
hist(x, data = NULL, subset = 1, include = 1:(x@K),
  histogram = TRUE, labels = TRUE, ylab = "Density", main = NULL,
  col = NULL, pch = 20, cex = 0.6, ...)

## S3 method for class 'flowClustList'
hist(x, ...)
```

Arguments

x	Object returned from [flowClust](#) or from running filter on a flowFrame object.
data	A numeric vector, matrix, data frame of observations, or object of class flowFrame. This is the object on which flowClust or filter was performed.
subset	An integer indicating which variable is selected for the plot. Alternatively, a character string containing the name of the variable is allowed if x@varNames is not NULL.
include	A numeric vector specifying which clusters are shown on the plot. By default, all clusters are included.
histogram	A logical value indicating whether a histogram of the actual data is made in addition to the density plot or not.

labels	A logical value indicating whether information about cluster assignment is shown or not.
ylab	Labels for the x- and y-axes respectively.
main	Title of the plot.
col	Colors of the plotting characters displaying the cluster assignment (if labels is TRUE). If NULL (default), it will be determined automatically.
pch	Plotting character used to show the cluster assignment.
cex	Size of the plotting character showing the cluster assignment.
...	other arguments
	xlim The range of x-values for the plot. If NULL, the data range will be used.
	ylim The range of y-values for the plot. If NULL, an optimal range will be determined automatically.
	breaks Content to be passed to the breaks argument of the generic hist function, if histogram is TRUE. Default is 50, meaning that 50 vertical bars with equal binwidths will be drawn.
	... Further arguments passed to curve (and also hist
	if histogram is TRUE).

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

flowClust, plot, density

Map,flowClust-method   *Cluster Assignment Based on Clustering Results*

Description

This method performs cluster assignment according to the posterior probabilities of clustering memberships resulted from the clustering (filtering) operations. Outliers identified will be left unassigned by default.

Usage

```
Map(f, ...)

## S4 method for signature 'flowClust'
Map(f, rm.outliers = TRUE, ...)

## S4 method for signature 'flowClustList'
Map(f, rm.outliers = TRUE, ...)
```

Arguments

f	Object returned from [flowClust](#) or [filter](#).
rm.outliers	A logical value indicating whether outliers will be left unassigned or not.
...	Further arguments to be passed to or from other methods.

Value

A numeric vector of size N (the number of observations) indicating to which cluster each observation is assigned. Unassigned observations will be labelled as NA.

Note

Even if rm.outliers is set to FALSE, NA may still appear in the resultant vector due to the filtered observations; see the descriptions about the min.count, max.count, min and max arguments of [flowClust](#).

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

[flowClust](#), [filter](#), [posterior](#)

---

mkPrior   *Generate a flowClust prior specification*

Description

Generate a flowClust prior specification from gates and data

Usage

```
mkPrior(gate, data, nu0, Omega0, ...)

## S4 method for signature 'polygonGate,flowFrame,numeric,matrix'
mkPrior(gate, data, nu0,
  Omega0)

## S4 method for signature 'rectangleGate,flowFrame,numeric,matrix'
mkPrior(gate, data, nu0,
  Omega0)

## S4 method for signature 'rectangleGate,flowFrame,missing,missing'
mkPrior(gate, data,
  nu0 = NA, Omega0 = NA)
```

```
## S4 method for signature 'polygonGate,flowFrame,missing,missing'
mkPrior(gate, data,
  nu0 = NA, Omega0 = NA)

## S4 method for signature 'list,flowSet,missing,missing'
mkPrior(gate, data, nu0 = NA, Omega0,
  model.cov = "full", model.means = "full")

## S4 method for signature 'missing,flowSet,ANY,missing'
mkPrior(gate, data, nu0 = NA, Omega0,
  model.cov = "full", model.means = "full")

## S4 method for signature 'missing,flowFrame,missing,missing'
mkPrior(gate, data, nu0, Omega0)

## S4 method for signature 'list,flowSet,ANY,missing'
mkPrior(gate, data, nu0 = NA, Omega0,
  model.cov = "full", model.means = "full")
```

Arguments

gate   A list of flowCore gates. The gates should represent the SAME population gated
                across multiple samples.

data   A flowSet of the same size as the number of gates above. Each flowFrame in the
                flowSet should contain the events representing the population in its correspond-
                ing gate. i.e. it should be the gated data.

nu0   The nu0 hyperparameter. For estimation from data, it should be nu0=NA.

Omega0   The Omega0 hyperparameter. For estimation from data it can be missing.

model.cov, model.means
                model names used for cov and means. one of c("full","DE","DU"). "full" is the
                default.

...   Not currently used.

Details

Construct a prior specification. Generally not called by the user.

Value

Return values depend on the specific method called. Not meant for user consumption.

Author(s)

Greg Finak <gfinak@fhcrc.org>

References

<http://www.rglab.org>

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
```

---

peakMatch   *Function to match peaks across samples*

Description

Uses the hungarian algorithm to match peaks across samples, one at a time using a template sample.

Usage

```
peakMatch(peaks, target.index, max.fill = 1e+12)
```

Arguments

peaks	is the matrix of peaks in the columns and samples in the rows
target.index	is the index of the template sample.
max.fill	is the value to substitute for NAs in the distance matrix. Should be very large, but if too large, will overflow and give an incorrect matching

---

plot,flowClust-method   *Scatterplot of Clustering Results*

Description

This method generates scatterplot revealing the cluster assignment, cluster boundaries according to the specified percentile as well as supplemental information like outliers or filtered observations.

Usage

```
plot(x, y, ...)

## S4 method for signature 'flowClust,missing'
plot(x, data, subset = c(1, 2), ellipse = T,
  show.outliers = T, show.rm = F, include = 1:(x@K), main = NULL,
  grayscale = F, col = (if (grayscale) gray(1/4) else 2:(length(include) +
  1)), pch = ".", cex = 0.6, col.outliers = gray(3/4),
  pch.outliers = ".", cex.outliers = cex, col.rm = 1, pch.rm = 1,
  cex.rm = 0.6, ecol = 1, elty = 1, level = NULL, u.cutoff = NULL,
  z.cutoff = NULL, npoints = 100, add = F, ...)

## S4 method for signature 'flowClustList,missing'
plot(x, data, subset = c(1, 2),
```

```
ellipse = T, show.outliers = T, show.rm = F, include = 1:(x@K),
main = NULL, grayscale = F, col = (if (grayscale) gray(1/4) else
2:(length(include) + 1)), pch = ".", cex = 0.6,
col.outliers = gray(3/4), pch.outliers = ".", cex.outliers = cex,
col.rm = 1, pch.rm = 1, cex.rm = 0.6, ecol = 1, elty = 1,
level = NULL, u.cutoff = NULL, z.cutoff = NULL, npoints = 501,
add = F, ...)
```

Arguments

x	Object returned from [flowClust]().
y	missing
data	A matrix, data frame of observations, or object of class flowFrame. This is the object on which flowClust was performed.
subset	A numeric vector of length two indicating which two variables are selected for the scatterplot. Alternatively, a character vector containing the names of the two variables is allowed if x@varNames is not NULL.
ellipse	A logical value indicating whether the cluster boundary is to be drawn or not. If TRUE, the boundary will be drawn according to the level specified by level or cutoff.
show.outliers	A logical value indicating whether outliers will be explicitly shown or not.
show.rm	A logical value indicating whether filtered observations will be shown or not.
include	A numeric vector specifying which clusters will be shown on the plot. By default, all clusters are included.
main	Title of the plot.
grayscale	A logical value specifying if a grayscale plot is desired. This argument takes effect only if the default values of relevant graphical arguments are taken.
col	Color(s) of the plotting characters. May specify a different color for each cluster.
pch	Plotting character(s) of the plotting characters. May specify a different character for each cluster.
cex	Size of the plotting characters. May specify a different size for each cluster.
col.outliers	Color of the plotting characters denoting outliers.
pch.outliers	Plotting character(s) used to denote outliers. May specify a different character for each cluster.
cex.outliers	Size of the plotting characters used to denote outliers. May specify a different size for each cluster.
col.rm	Color of the plotting characters denoting filtered observations.
pch.rm	Plotting character used to denote filtered observations.
cex.rm	Size of the plotting character used to denote filtered observations.
ecol	Color(s) of the lines representing the cluster boundaries. May specify a different color for each cluster.
elty	Line type(s) drawing the cluster boundaries. May specify a different line type for each cluster.
level, u.cutoff, z.cutoff	
	These three optional arguments specify the rule used to identify outliers. By default, all of them are left unspecified, meaning that the rule stated in x@ruleOutliers will be taken. Otherwise, these arguments will be passed to [ruleOutliers]().

npoints	The number of points used to draw each cluster boundary.
add	A logical value. If `TRUE`, add to the current plot.
...	Further graphical parameters passed to the generic function `plot`.

Note

The cluster boundaries need not be elliptical since Box-Cox transformation has been performed.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

[flowClust](#)

---

plot,flowDens-method   *Contour or Image Plot of Clustering Results*

Description

This method makes use of the `flowDens` object returned by [density](#) to generate a contour or image plot.

Usage

```
## S4 method for signature 'flowDens,missing'
plot(x, type = c("contour", "image"),
  nlevels = 30, scale = c("raw", "log", "sqrt"), color = c("rainbow",
  "heat.colors", "terrain.colors", "topo.colors", "cm.colors", "gray"),
  xlab = colnames(x@dx), ylab = colnames(x@dy), ...)
```

Arguments

x	The `flowDens` object returned from [density](#).
type	Either `"contour"` or `"image"` to specify the type of plot desired.
nlevels	An integer to specify the number of contour levels or colors shown in the plot.
scale	If `"log"`, the logarithm of the density values will be used to generate the plot; similar interpretation holds for `"sqrt"`. The use of a `log` or `sqrt` elicits more information about low density regions.
color	A string containing the name of the function used to generate the desired list of colors.
xlab, ylab	Labels for the x- and y-axes respectively.
...	Other arguments to be passed to `contour` or `image`, for example, `drawlabels` and `add`. Once an image plot is generated, users may impose a contour plot on it by calling this function with an additional argument add=TRUE.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

[flowClust](), [density]()

---

plot,flowFrame,tmixFilterResult-method
                     Scatterplot / 1-D Density Plot of Filtering (Clustering) Results

---

Description

Depending on the dimensions specified, this method generates either a scatterplot or a one-dimensional density plot (histogram) based on the robust model-based clustering results.

Usage

```
## S4 method for signature 'flowFrame,tmixFilterResult'
plot(x, y, z = NULL, ...)

## S4 method for signature 'flowFrame,tmixFilterResultList'
plot(x, y, z = NULL, ...)
```

Arguments

x	Object of class flowFrame. This is the data object on which [filter]() was performed.
y	Object of class tmixFilterResult or tmixFilterResultList returned from running [filter]().
z	A character vector of length one or two containing the name(s) of the variable(s) selected for the plot. If it is of length two, a scatterplot will be generated. If it is of length one, a 1-D density plot will be made. If it is unspecified, the first one/two variable(s) listed in y@varNames will be used.
...	All optional arguments passed to the [plot]() or [hist]() method with signature 'flowClust'. Note that arguments x, data and subset have already been provided by y, x and z above respectively.

Note

This plot method is designed such that it resembles the argument list of the plot method defined in the **flowCore** package. The actual implementation is done through the [plot]() or [hist]() method defined for a flowClust object.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

[filter](), [plot](), [hist]()

---

plotPrior	*Plots a flowClust prior over some data.*

Description

Plots a flowClust prior overlaid on data.

Usage

```
plotPrior(data, prior, dims = NULL, ...)
```

Arguments

data	On object of class "flowFrame". The data to be plotted.
prior	An object of class "flowClustPrior", or "flowClustPriorList", returned by a call to mkPrior.
dims	A character vector of the dimensions to be included in the plot. The dimension names should match column names in the prior and in the flowFrame.
...	Additional arguments to plotting functions, such as smooth=TRUE/FALSE

Details

Generates a plot of a "flowClustPrior" or "flowClustPriorList" object overlaid on some data. Plots the prior means (Mu0), prior covariance of the means (Omega0), and prior sample covariance (Lambda0).

Value

Silently returns zero.

Author(s)

Greg Finak <gfinak@fhcrc.org>

---

rbox   *Reverse Box-Cox Transformation*

Description

This function performs back transformation on Box-Cox transformed data.

Usage

```
rbox(data, lambda)
```

Arguments

data   A numeric vector, matrix or data frame of observations.

lambda   The Box-Cox transformation applied which results in the inputted data matrix.

Value

A numeric vector, matrix or data frame of the same dimension as data is returned.

Note

Please refer to the documentation for box for details about the Box-Cox transformation in use.

See Also

box

---

rituximab   *The Rituximab Dataset*

Description

A flow cytometry dataset produced in a drug-screening project to identify agents that would enhance the anti-lymphoma activity of Rituximab, a therapeutic monoclonal antibody. Cells were stained with anti-BrdU FITC and the DNA binding dye 7-AAD.

Format

An object of class flowFrame with 1545 cells (rows) and the following eight variables (columns):

FSC.H FSC-Height

SSC.H Side Scatter

FL1.H Anti-BrdU FITC

FL2.H Channel not used

FL3.H 7 AAD

FL1.A Channel not used

FL1.W Channel not used

Time Time

Source

Gasparetto, M., Gentry, T., Sebti, S., O'Bryan, E., Nimmanapalli, R., Blaskovich, M. A., Bhalla, K., Rizzieri, D., Haaland, P., Dunne, J. and Smith, C. (2004) Identification of compounds that enhance the anti-lymphoma activity of rituximab using flow cytometric high-content screening. *J. Immunol. Methods* **292**, 59-71.

---

ruleOutliers,flowClust-method

Showing or Modifying the Rule used to Identify Outliers

Description

This method shows or modifies the rule used to identify outliers.

Usage

```
ruleOutliers(object)

## S4 method for signature 'flowClust'
ruleOutliers(object)

## S4 method for signature 'flowClustList'
ruleOutliers(object)

ruleOutliers(object) <- value

## S4 replacement method for signature 'flowClust,list'
ruleOutliers(object) <- value
```

Arguments

object	Object returned from [flowClust](#) or [filter](#).
value	A list object with one or more of the following named elements: `level`, `u.cutoff` and `z.cutoff`. Their interpretations are the same as those of the corresponding arguments in the [flowClust](#) function. Note that when both `level` and `u.cutoff` are missing, the rule set by the original value of `level` or `u.cutoff` will be unchanged rather than removed. Likewise, when `z.cutoff` is missing, the rule set by the original value of `z.cutoff` will be retained.

Value

The replacement method modifies `object@ruleOutliers` (or `object[[k]]@ruleOutliers` if `object` is of class `flowClustList` or `tmixFilterResultList`) AND updates the logical vector `object@flagOutliers` (or `object[[k]]@ruleOutliers`) according to the new rule.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

[flowClust](), [filter]()

---

show,flowClust-method   *Show Method for flowClust / tmixFilterResult Object*

Description

This method lists out the slots contained in a flowClust object.

Usage

```
## S4 method for signature 'flowClust'
show(object)

## S4 method for signature 'flowClustList'
show(object)

## S4 method for signature 'tmixFilter'
show(object)

## S4 method for signature 'tmixFilterResult'
show(object)

## S4 method for signature 'tmixFilterResultList'
show(object)
```

Arguments

object   Object returned from [flowClust]() or [filter]().

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

See Also

[flowClust](), [filter](), [summary]()

SimulateMixture	*Random Generation from a t Mixture Model with Box-Cox Transformation*

Description

This function can be used to generate a sample from a multivariate t mixture model with Box-Cox transformation.

Usage

```
SimulateMixture(N, w, mu, sigma, nu = 4, lambda)
```

Arguments

N	The number of observations.
w	A vector of length K, containing the K cluster proportions.
mu	A matrix of size $K \times P$, where K is the number of clusters and P is the dimension, containing the K mean vectors.
sigma	An array of dimension $K \times P \times P$, containing the K covariance matrices.
nu	The degrees of freedom used for the t distribution.
lambda	The Box-Cox transformation parameter. If missing, the conventional t distribution without transformation will be used.

Value

A matrix of size $N \times P$.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

See Also

flowClust

Examples

```
library(flowCore)
library(flowClust)
### Number of components
K <- 5
### Dimension
p <- 2
### Number of observations
n <- 200
Mu <- matrix(runif(K*p, 0, 20), K, p)
Sigma <- array(0, c(K, p, p))

for (k in 1:K)
{
    Sigma[k,,][outer(1:p, 1:p, ">")] <- runif(p*(p-1)/2,-.1,.1)
```

```
    diag(Sigma[k,,]) <- runif(p,0,1)
    ### Make sigma positive definite
    Sigma[k,,] <- Sigma[k,,] %*% t(Sigma[k,,])
}

### Generate the weights
w <- rgamma(K,10,1)
w <- w/sum(w)

y <- SimulateMixture(n, w, Mu, Sigma, nu=4)
```

---

split,flowClust-method

Splitting Data Based on Clustering Results

Description

This method splits data according to results of the clustering (filtering) operation. Outliers identified will be removed by default.

Usage

```
split(x, f, drop = FALSE, ...)

## S4 method for signature 'data.frame,flowClust'
split(x, f, drop = FALSE,
  population = NULL, split = NULL, rm.outliers = TRUE, ...)

## S4 method for signature 'matrix,flowClust'
split(x, f, drop = FALSE, population = NULL,
  split = NULL, rm.outliers = TRUE, ...)

## S4 method for signature 'vector,flowClust'
split(x, f, drop = FALSE, population = NULL,
  split = NULL, rm.outliers = TRUE, ...)

## S4 method for signature 'flowFrame,flowClust'
split(x, f, drop = FALSE, population = NULL,
  split = NULL, rm.outliers = TRUE, ...)

## S4 method for signature 'flowFrame,tmixFilterResult'
split(x, f, drop = FALSE,
  population = NULL, split = NULL, rm.outliers = TRUE, ...)

## S4 method for signature 'flowFrame,flowClustList'
split(x, f, drop = FALSE,
  population = NULL, split = NULL, rm.outliers = TRUE, ...)

## S4 method for signature 'data.frame,flowClustList'
split(x, f, drop = FALSE,
  population = NULL, split = NULL, rm.outliers = TRUE, ...)
```

```
## S4 method for signature 'matrix,flowClustList'
split(x, f, drop = FALSE,
  population = NULL, split = NULL, rm.outliers = TRUE, ...)

## S4 method for signature 'vector,flowClustList'
split(x, f, drop = FALSE,
  population = NULL, split = NULL, rm.outliers = TRUE, ...)

## S4 method for signature 'flowFrame,tmixFilterResultList'
split(x, f, drop = FALSE,
  population = NULL, split = NULL, rm.outliers = TRUE, ...)
```

Arguments

x	A numeric vector, matrix, data frame of observations, or object of class `flowFrame`. This is the object on which [flowClust](#) or [filter](#) was performed.
f	Object returned from `flowClust` or `filter`.
drop	A logical value indicating whether to coerce a column matrix into a vector, if applicable. Default is `FALSE`, meaning that a single-column matrix will be retained.
population	An optional argument which specifies how to split the data. If specified, it takes a list object with named or unnamed elements each of which is a numeric vector specifying which clusters are included. If this argument is left unspecified, the data object will be split into K subsets each of which is formed by one out of the K clusters used to model the data. See examples for more details.
split	This argument is deprecated. Should use `population` instead.
rm.outliers	A logical value indicating whether outliers are removed or not.
...	Further arguments to be passed to or from other methods.

Value

A list object with elements each of which is a subset of x and also retains the same class as x. If the `split` argument is specified with a list of named elements, those names will be used to name the corresponding elements in the resultant list object.

Usage

split(x, f, drop=FALSE, population=NULL, split=NULL, rm.outliers=TRUE, . . .)

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

[Subset](#), [flowClust](#), [filter](#)

Subset,flowClust-method

Subsetting Data Based on Clustering Results

Description

This method returns a subset of data upon the removal of outliers identified from the clustering (filtering) operations.

Usage

```
## S4 method for signature 'flowFrame,flowClust'
Subset(x, subset, ...)

## S4 method for signature 'flowFrame,tmixFilterResult'
Subset(x, subset, ...)

## S4 method for signature 'data.frame,flowClust'
Subset(x, subset, ...)

## S4 method for signature 'matrix,flowClust'
Subset(x, subset, ...)

## S4 method for signature 'vector,flowClust'
Subset(x, subset, ...)

## S4 method for signature 'ANY,flowClustList'
Subset(x, subset, ...)

## S4 method for signature 'flowFrame,tmixFilterResultList'
Subset(x, subset, ...)
```

Arguments

x	A numeric vector, matrix, data frame of observations, or object of class flowFrame. This is the object on which [flowClust](#) or [filter](#) was performed.
subset	Object returned from flowClust or filter.
...	Further arguments to be passed to or from other methods.

Value

An object which is a subset of x. It also retains the same class as x.

Usage

Subset(x, subset, . . .)

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

split, flowClust, filter

---

summary,flowClust-method

Summary Method for flowClust Object

Description

This method prints out various characteristics of the model fitted via robust model-based clustering.

Usage

```
summary(object, ...)

## S4 method for signature 'flowClust'
summary(object)

## S4 method for signature 'flowClustList'
summary(object)
```

Arguments

object   Object returned from flowClust or from filter.

...   not used

Details

Various characteristics of the fitted model will be given under the following five categories: Experiment Information, Clustering Summary, Transformation Parameter, Information Criteria, and Data Quality. Under Data Quality, information about data filtering, outliers, and uncertainty is given.

Author(s)

Raphael Gottardo <<raph@stat.ubc.ca>>, Kenneth Lo <<c.lo@stat.ubc.ca>>

See Also

flowClust, filter, show

tmixFilter-class   *Creating Filters and Filtering Flow Cytometry Data*

Description

The `tmixFilter` function creates a filter object which is then passed to the `filter` method that performs filtering on a `flowFrame` object. This method pair is provided to let **flowClust** integrate with the **flowCore** package.

Usage

```
tmixFilter(filterId = "tmixFilter", parameters = "", ...)

## S4 method for signature 'ANY,flowClust'
x %in% table

## S4 method for signature 'flowFrame,tmixFilterResult'
x %in% table

## S4 method for signature 'flowFrame,tmixFilter'
x %in% table

## S4 method for signature 'ANY,tmixFilterResult'
x %in% table

## S4 method for signature 'ANY,flowClustList'
x %in% table

## S4 method for signature 'ANY,tmixFilterResultList'
x %in% table

## S4 method for signature 'flowFrame,flowClust,ANY,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'flowFrame,tmixFilterResult,ANY,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'flowFrame,flowClustList,ANY,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'flowFrame,tmixFilterResultList,ANY,ANY'
x[i, j, ...,
  drop = FALSE]

## S4 method for signature 'tmixFilterResultList,ANY'
x[[i, j, ..., exact = TRUE]]

## S4 method for signature 'tmixFilterResultList'
length(x)

## S4 method for signature 'tmixFilterResult,tmixFilter'
```

```
summarizeFilter(result, filter)
```

Arguments

filterId	A character string that identifies the filter created.
parameters	A character vector specifying the variables to be used in filtering. When it is left unspecified, all the variables of the flowFrame object are used when running filter. Note that its content will be passed to the varNames argument of flowClust when running filter.
x	flowFrame
table	tmixFilterResult
i	tmixFilterResult or tmixFilterResultList
j, drop, exact	not used
result	tmixFilterResult
filter	tmixFilter
...	Other arguments passed to the flowClust function when running filter, namely, expName, K, B, tol, nu, lambda, nu.est, trans, min.count, max.count, min, max, level, u.cutoff, z.cutoff, randomStart, B.init, tol.init, seed and criterion. All arguments are optional except K that specifies the number of clusters.

The tmixFilter function returns an object of class tmixFilter that stores all the settings required for performing the filter operations.

The filter method is defined in package flowCore and returns an object of class tmixFilterResult (or tmixFilterResultList if filter has a length >1) that stores the filtering results.

The tmixFilter function returns an object of class tmixFilter that extends the virtual parent filter class in the **flowCore** package. Hence, the filter operators, namely, &, |, ! and subset, also work for the tmixFilter class.

If filter is of length 1, the filter method returns an object of class tmixFilterResult. This class extends both the multipleFilterResult class (in the **flowCore** package) and the flowClust class. Operations defined for the multipleFilterResult class, like %in%, Subset and split, also work for the tmixFilterResult class. Likewise, methods or functions designed to retrieve filtering (clustering) information from a flowClust object can also be applied on a tmixFilterResult object. These include criterion, ruleOutliers, ruleOutliers<-, Map, posterior, importance, uncertainty and getEstimates. Various functionalities for plotting the filtering results are also available (see the links below).

If filter has a length >1, the function returns an object of class tmixFilterResultList. This class extends both the flowClustList class and the multipleFilterResult class. Note that when a tmixFilterResultList object is used in place of a tmixFilterResult object, in most cases the list element corresponding to the best model will be extracted and passed to the method/function call.

References

Lo, K., Brinkman, R. R. and Gottardo, R. (2008) Automated Gating of Flow Cytometry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332.

See Also

flowClust, summary, plot, density, hist, Subset, split, ruleOutliers, Map

Examples

```
### The example below largely resembles the one in the flowClust
### man page.  The main purpose here is to demonstrate how the
### entire cluster analysis can be done in a fashion highly
### integrated into flowCore.


data(rituximab)
library(flowCore)

### create a filter object
s1filter <- tmixFilter("s1", c("FSC.H", "SSC.H"), K=1)
### cluster the data using FSC.H and SSC.H
res1 <- filter(rituximab, s1filter)

### remove outliers before proceeding to the second stage
# %in% operator returns a logical vector indicating whether each
# of the observations lies inside the gate or not
rituximab2 <- rituximab[rituximab %in% res1,]
# a shorthand for the above line
rituximab2 <- rituximab[res1,]
# this can also be done using the Subset method
rituximab2 <- Subset(rituximab, res1)

### cluster the data using FL1.H and FL3.H (with 3 clusters)
s2filter <- tmixFilter("s2", c("FL1.H", "FL3.H"), K=3)
res2 <- filter(rituximab2, s2filter)

show(s2filter)
show(res2)
summary(res2)

# to demonstrate the use of the split method
split(rituximab2, res2)
split(rituximab2, res2, population=list(sc1=c(1,2), sc2=3))

# to show the cluster assignment of observations
table(Map(res2))

# to show the cluster centres (i.e., the mean parameter estimates
# transformed back to the original scale) and proportions
getEstimates(res2)

### demonstrate the use of various plotting methods
# a scatterplot
plot(rituximab2, res2, level=0.8)
plot(rituximab2, res2, level=0.8, include=c(1,2), grayscale=TRUE,
    pch.outliers=2)
# a contour / image plot
res2.den <- density(res2, data=rituximab2)
plot(res2.den)
plot(res2.den, scale="sqrt", drawlabels=FALSE)
plot(res2.den, type="image", nlevels=100)
plot(density(res2, include=c(1,2), from=c(0,0), to=c(400,600)))
# a histogram (1-D density) plot
```

```
plot(rituximab2, res2, "FL1.H")

### to demonstrate the use of the ruleOutliers method
summary(res2)
# change the rule to call outliers
ruleOutliers(res2) <- list(level=0.95)
# augmented cluster boundaries lead to fewer outliers
summary(res2)

# the following line illustrates how to select a subset of data
# to perform cluster analysis through the min and max arguments;
# also note the use of level to specify a rule to call outliers
# other than the default
s2t <- tmixFilter("s2t", c("FL1.H", "FL3.H"), K=3, B=100,
    min=c(0,0), max=c(400,800), level=0.95, z.cutoff=0.5)
filter(rituximab2, s2t)
```

Index