flowMerge

April 20, 2011

`checkForRemoteErrors` *Check output of snow clusters for errors*

Description

Overrides the snow `checkForRemoteErrors` function. Try errors are returned when cluster nodes produce errors, rather than completely aborting the computation. Not meant to be called by the user.

Usage

```
checkForRemoteErrors(val)
```

Arguments

`val`	The result returned from an individual cluster node.

Details

This function is meant to be called internally, but must be exported so that it can hide the native `checkForRemoteErrors` function in the snow package.

Value

The result from the snow cluster node, or an object of type `try-error` if there was an error.

Author(s)

Greg Finak «`greg.finak@ircm.qc.ca`»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data; Advances in Bioinformatics (To Appear)

See Also

checkForRemoteErrors

`fitPiecewiseLinreg-methods` *Methods for fitPiecewiseLinreg in flowMerge package*

Description

Methods for the function `fitPiecewiseLinreg` in the package `flowMerge`

Methods

x = "list" A list of `flowMerge` objectes derived from a call to the `merge` function.

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data; Advances in Bioinformatics (To Appear)

`fitPiecewiseLinreg` *Fit Piecewise Linear Regression for a list of flowMerge Objects*

Description

Fits a two–component piecewise linear regression to the entropy vs number of clusters for a list of merged cluster solutions.

Usage

```
fitPiecewiseLinreg(x, plot=FALSE, normalized=TRUE, ...)
```

Arguments

`x`	A "list" of `flowMerge` objects for `1` through `K` clusters derived from a single max BIC `flowObj` or `flowClust` object.
`plot`	A `logical` indicating whether to plot the fit or not. Default is FALSE.
`normalized`	A `logical` indicating whether the merging should be done using the normalized or unnormalized entropy
`...`	Additional arguments not currently used.

Details

An S4 method that takes a list of `flowMerge` objects output by the `merge` method, extracts the entropy and fits a piecwise linear regression to the entropy vs number of clusters in order to find the postion of the changepoint. The location of the changepoint corresponds to the optimal merged cluster solution. The piecewise linear regression now is fitted to the entropy vs cumulative sum of merged observations at each number of clusters. This normalizes the change in entropy for the number of data points as described in Baudry et al.

Value

An `integer` value corresponding to the position of the changepoint.

Author(s)

Greg Finak «`greg.finak@ircm.qc.ca`»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data; Advances in Bioinformatics (To Appear)

See Also

merge

Examples

```
data(rituximab)
data(RituximabFlowClustFit)
o<-flowObj(flowClust.res[[which.max(BIC(flowClust.res))]],rituximab);
m<-merge(o)
i<-fitPiecewiseLinreg(m);
```

`flagOutliers-methods` *Methods to update the flagOutliers slot in a flowMerge object.*

Description

Methods that update the `flagOutliers` slot in a flowMerge object so that they reflec the outliers in the new merged clustering. This is an internal function, not meant for user consumption. It is called from within the `merge` method.

Methods

object = "flowMerge" Update the `flagOutliers` slot for an object of type `flowMerge`

`flagOutliers` *Update the flagOutliers slot in a flowMerge object*

Description

Update the flagOutliers slot in a flowMerge object. This method is internal and called automatically from within the merging code.

Usage

```
flagOutliers(object,...)
```

Arguments

`object`	An object of type `flowMerge`
`...`	Additional arguments, currently unused.

`flowClust-class` *Class "flowClust"*

Description

The basic class used in the **flowClust** package

Slots

`BIC`: The Bayesian Information Criterion for the fitted mixture model.

`ICL`: The Integrated Completed Likelihood for the fitted mixture model.

`expName`: A `"character"` vector with the name of the experiment.

`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'.

`varNames`: A `"character"` vector containing the variable names of the data channels. Usually named something like `"FL1 Log"`, `"FL2 Log"`, `"FS Lin"`, code"SS Lin", rather than the names of the immunohistochemical markers.

`K`: The number of components in the model.

`w`: A `"vector"` containing the weights or proportions associated with each mixture component.

`mu`: A `"matrix"` of K x p dimensions containing the means of the mixture components in the model.

`sigma`: An `"array"` of K x p x p containing the sigma parameters for the mixture components in the model.

`lambda`: A `"numeric"` with the estimated parameter `lambda` used in the model.

`logLike`: The log-likelihood of the fitted mixture model.

`nu`: A `"numeric"`. The degrees of freedom used in the model.

`rm.min`: Number of points filtered from below.

`rm.max`: Number of points filtered from above.

- `z`: A `"matrix"` of size N x K containing the posterior probabilities of cluster memberships. The probabilities in each row sum up to one.
- `u`: A `"matrix"` of size N x 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' is 'FALSE'). 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

Methods

See the **flowClust** package.

Note

Please see the `flowClust` package documentation for a detailed description of the slots in a `flowClust` object. This class is the parent class for `flowMerge` and `flowObj` classes used in the `flowMerge` package.

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 Cytomet ry Data via Robust Model-based Clustering. *Cytometry A* **73**, 321-332 .

See Also

flowClust

`flowClust.res` *A flowClust model fitted to the rituximab data for 1:10 clusters.*

Description

The Rituximab data set accessible via `data(rituximab)` in the `flowClust` package fitted to a flowClust model containing from one to ten components. The results are in the object `flowClust.res`.

Usage

```
data(RituximabFlowClustFit)
```

Format

The format is: `flowClust.res` is a `flowClustList`, where each element of the list is a `flowClust` model of the `rituximab` data, for `K=1` through `K=10` components, respectively. The structure of `flowClustList` and `flowClust` can be found in the corresponding documentation of the `flowClust` package. The format of the `rituximab` data is found in the documentation for that data set.

Details

The models have been precomputed for use in `flowMerge` examples to save computation time. `flowClust` was called on the rituximab data to generate these models with the following command: `flowClust.res<-flowClust(rituximab,K=1:10,B=1000,B.init=100,tol=1e-5,tol.init=1e-2,nu=4,randomStart=50,trans=1,nu.est=1).`

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.

Examples

```
data(RituximabFlowClustFit)
summary(flowClust.res);
```

`flowMerge-class` *Class "flowMerge"*

Description

A class to represent flowMerge objects

Objects from the Class

The object unites the flowMerge model output and the data being modeled and contains additional slots for various characteristics of a merged cluster solution, including the entropy of clustering.

Slots

`merged`**:** A list containing the cluster numbers merged at each iteration of the flowMerge algorithm, in order to arrive at the current solution

`entropy`**:** The entropy of clustering of the current solution.

`DATA`**:** An environment whose first element contains the flowFrame with the data modeled by this flowMerge object

`expName`**:** See the `flowClust` package for details `varNames`: See the `flowClust` package for details

`K`: The number of clusters in the merged solution. See the `flowClust` package for details

`w`: The proportions for each component in the merged solution. See the `flowClust` package for details

`mu`: The means of the components in the merged solution. See the `flowClust` package for details

`sigma`: The covraiances of the components in the merged solution. See the `flowClust` package for details

`lambda`: See the `flowClust` package for details

`nu`: See the `flowClust` package for details

`z`: See the `flowClust` package for details

`u`: The uncertainties for each data point.

`label`: See the `flowClust` package for details

`uncertainty`: See the `flowClust` package for details

`ruleOutliers`: See the `flowClust` package for details

`flagOutliers`: See the `flowClust` package for details

`rm.min`: See the `flowClust` package for details

`rm.max`: See the `flowClust` package for details

`logLike`: See the `flowClust` package for details

`BIC`: See the `flowClust` package for details

`ICL`: See the `flowClust` package for details

Extends

Class `"flowObj"`, directly. Class `"flowClust"`, by class "flowObj", distance 2.

Methods

getData `signature(obj = "flowMerge")`: Retrieves the `flowFrame` in the `DATA` environment slot.

plot `signature(x = "flowMerge", y = "missing")`: Plots the clusters in this object.

summary `signature(x="flowMerge")`: Prints a summary of the object.

show `signature(x="flowMerge")`: Prints information about the object.

Author(s)

Greg Finak «`greg.finak@ircm.qc.ca`»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data (Submitted)

See Also

flowObj-class

flowMerge-package *Merging of mixture components for automated gating of flow cytometry data.*

Description

Merges mixture components from the **flowClust** framework based on the entropy of clustering and provides a simple representation of complicated, non-convex cell populations.

Details

Package:	flowMerge
Type:	Package
Version:	0.4.1
Date:	2009-09-07
License:	Artistic-2.0
LazyLoad:	yes
Depends:	methods

High density, non-convex cell populations in flow cytometry data often require multiple mixture components for a good model fit. The components are often overlapping, resulting in a complicated representation of individual cell populations. flowMerge merges overlapping mixture components (based on the max BIC `flowClust` model fit) in an iterative manner based on an entropy criterion, allowing these cell populations to be represented by individual mixture components while retaining the good model fitting properties of the BIC solution. Estimates of the number of clusters from a `flowMerge` model more accurately represent the "true" number of cell populations in the data. Running `flowMerge` is relatively straightforward. A `flowClust` object is converted to a `flowObj` object, which groups the model and the data (a `flowFrame`) into a single object. This is done by a call to `flowObj(model, data)` with a call to `merge`, which takes a `flowObj` object. The algorithm may be run in parallel on a multi-core machine or a networked cluster of machines. It uses the functionality in the `snow` package to achieve this. Parallelized calls to `flowClust` are available via the `pFlowClust` and `pFlowMerge` functions.

`flowMerge` has functionality to automatically select the "correct" number of clusters by fitting a piecewise linear model to the entropy of clustering vs number of clusters, and locating the position of the changepoint. The piecewise linear model fitting is invoked by a call to `fitPiecewiseLinreg`, which returns the location of the changepoint.

Author(s)

Greg Finak «`greg.finak@ircm.qc.ca`», Raphael Gottardo «`raphael.gottardo@ircm.qc.ca`»

Maintainer: Greg Finak «`greg.finak@ircm.qc.ca`»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data; Advances in Bioinformatics (To Appear)

See Also

`flowClust`,`flowObj`,`pFlowMerge`,`pFlowClust`,`fitPiecewiseLinreg`,`merge`,`getData`,`link{pl`

Examples

```
data(rituximab)
data(RituximabFlowClustFit)
o<-flowObj(flowClust.res[[which.max(flowMerge:::BIC(flowClust.res))]],rituximab);
m<-merge(o);
i<-fitPiecewiseLinreg(m);
m<-m[[i]];
plot(m,pch=20,level=0.9);
```

`flowObj-class` *Class "flowObj"*

Description

A class inheriting from `flowClust` that groups the model and data in a single object.

Objects from the Class

Objects can be created by calls of the form `new("flowObj", ...)`. Has a convenience method `flowObj(flowClustObj, flowFrameObj)` for creating instances of the class.

Slots

`DATA`: An `"environment"` that holds a pointer to the `flowFrame` data in position `[[1]]`.

`expName`: As described in the `flowClust` documentation

`varNames`: As described in the `flowClust` documentation

`K`: As described in the `flowClust` documentation

`w`: As described in the `flowClust` documentation

`mu`: As described in the `flowClust` documentation

`sigma`: As described in the `flowClust` documentation

`lambda`: As described in the `flowClust` documentation

`nu`: As described in the `flowClust` documentation

`z`: As described in the `flowClust` documentation

`u`: As described in the `flowClust` documentation

`label`: As described in the `flowClust` documentation

`uncertainty`: As described in the `flowClust` documentation

`ruleOutliers`: As described in the `flowClust` documentation

`flagOutliers`: As described in the `flowClust` documentation

`rm.min`: As described in the `flowClust` documentation

`rm.max`: As described in the `flowClust` documentation

`logLike`: As described in the `flowClust` documentation

`BIC`: As described in the `flowClust` documentation

`ICL`: As described in the `flowClust` documentation

Extends

Class "`flowClust`", directly.

Methods

getData `signature(obj = "flowObj")`: Retreives the contents of the `DATA` environment

merge `signature(x = "flowObj", y = "missing")`: the `flowMerge` algorithm is called via this function on objects of type `flowObj`.

plot `signature(x = "flowObj", y = "missing")`: A simplified plotting method. Does not require specification of the data since it is contained in the `flowObj` object. Takes most of the same parameters as `plot.flowClust`, except the `data` parameter

Author(s)

Greg Finak «`greg.finak@ircm.qc.ca`», Raphael Gottardo «`raphael.gottardo@ircm.qc.ca`»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data; Advances in Bioinformatics (To Appear)

See Also

flowMerge-class, flowObj

`flowObj` *Create a flowObj object from a flowClust and flowFrame object*

Description

Convenience method that creates a `flowObj` object from a `flowClust` and `flowFrame` object, so as to group the model and data together. Useful for high-throughput analysis where one may want to access the data to compute other statistics.

Usage

```
flowObj(flowC = NULL, flowF = NULL)
```

Arguments

`flowC`	A `flowClust` object representing the model fit
`flowF`	A `flowFrame` object on which the flowClust model is based.

Details

Calls the `new("flowObj",..)` constructor.

Value

An object of class `flowObj-class`

Author(s)

Greg Finak «greg.finak@ircm.qc.ca», Raphael Gottardo «raphael.gottardo@ircm.qc.ca»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data; Advances in Bioinformatics (To Appear)

See Also

flowObj-class

Examples

```
data(rituximab)
data(RituximabFlowClustFit)
o<-flowObj(flowClust.res[[which.max(flowMerge:::BIC(flowClust.res))]],rituximab);
m<-merge(o);
```

---

initPFlowMerge *Initialize a SNOW cluster for use with flowMerge*

Description

Initializes a snow cluster for use with flowMerge, ensures that the flowMerge library is loaded in all environments. Not meant to be called by the user

Usage

```
initPFlowMerge(cl)
```

Arguments

cl	A snow cluster

Details

A valid snow cluster.

Author(s)

Greg Finak «greg.finak@ircm.qc.ca»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data; Advances in Bioinformatics (To Appear)

See Also

pFlowMerge

map *Map matrix of probabilities to class assignments.*

Description

Traverse the rows of a matrix of probabilities of size n x k, where the n rows are samples, and the k columns are the probability of assignment of the sample to each of k classes. The most probable class assignment is selected for each row and a vector of classes is returned.

Usage

```
map(z, ...)
```

Arguments

`z`	A matrix of probabilities.
`...`	Additional arguments, not currently used.

Value

A vector of class assignments of lenght `n`.

Author(s)

Greg Finak <greg.finak@ircm.qc.ca>, Raphael Gottardo <raphael.gottardo@ircm.qc.ca>

Examples

```
z<-t(apply(t(replicate(100,rgamma(5,0.1,1))),1,function(x)x/sum(x)));
map(z);
```

mergeClusters2 *Cluster merging not meant to be called by the user*

Description

Internal function not meant to be called by the user.

Usage

```
mergeClusters2(object, a, b, data)
```

Arguments

`object`	Internal function not meant to be called by the user.
`a`	Internal function not meant to be called by the user.
`b`	Internal function not meant to be called by the user.
`data`	Internal function not meant to be called by the user.

Details

Internal function not meant to be called by the user.

Value

Internal function not meant to be called by the user.

Author(s)

Greg Finak «`greg.finak@ircm.qc.ca`»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data (Submitted)

See Also

merge

`mergeClusters` *Cluster merging not meant to be called by the user*

Description

Internal cluster merging function.

Usage

```
mergeClusters(object, data)
```

Arguments

`object`	not meant to be called by the user
`data`	not meant to be called by the user

Details

Not meant to be called by the user

Value

Not meant to be called by the user

Author(s)

Greg Finak «`greg.finak@ircm.qc.ca`»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data (Submitted)

See Also

merge

merge-methods *Merge mixture components*

Description

Merge mixture components in a `flowObj` derived from a `flowClust` result and a `flowFrame` using the cluster merging algorithm.

Value

An unnamed list of `flowMerge` objects with the `k`th element corresponding to the k-cluster merged solution.

Methods

x = "ANY", y = "ANY" The generic method. Should not be called.

x = "flowObj", y = "missing" The merge method for a `flowObj`.

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data (To Appear)

Examples

```
data(rituximab)
data(RituximabFlowClustFit)
o<-flowObj(flowClust.res[[which.max(flowMerge:::BIC(flowClust.res))]],rituximab);
m<-merge(o);
```

merge *Merge clusters in flow cytometry data*

Description

Merge the clusters in a flowClust solution using the cluster merging algorithm and entropy criterion.

Usage

```
merge(x,y,...)
```

Arguments

`x`	A `flowObj` object created from a `flowClust` object and a `flowFrame` using the `flowObj` constructor.
`y`	missing
`...`	Additional arguments, not currently used.

Details

Run the cluster merging algorithm on the max BIC solution from a call to `flowClust`.

Value

A list of unnamed `flowMerge` objects. The first element of the list corresponds to the 1–cluster merged solution. The second element corresponds to the 2–cluster merged solution, and so on.

Author(s)

Greg Finak «`greg.finak@ircm.qc.cq`»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data (Submitted)

See Also

flowClust,flowObj

Examples

```
data(rituximab)
data(RituximabFlowClustFit)
o<-flowObj(flowClust.res[[which.max(BIC(flowClust.res))]],rituximab)
m<-merge(o);
```

NENT *Extract the Normalized Entropy*

Description

Extracts the normalized entropy from a list of flowMerge objects.

Usage

```
NENT(x)
```

Arguments

`x` A list of flowMerge objects

Details

The normalized entropy is extracted from a flowMerge object by computing $\frac{E}{K*n}$ where E is the entropy, and K and n are the number of clusters and data points, respectively.

Value

Returns a vector of normalized entropy values for the flowMerge objects.

Warning

This function doesn't do enough error checking and will try to extract the entropy from a list of anything.

Author(s)

Greg Finak «greg.finak@ircm.qc.ca»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data; Advances in Bioinformatics (To Appear)

See Also

ENT,merge,flowMerge-class

Examples

```
data(RituximabFlowClustFit)
data(rituximab)
o<-flowObj(flowClust.res[[which.max(flowMerge:::BIC(flowClust.res))]],rituximab);
m<-merge(o);
flowMerge:::ENT(m);
flowMerge:::NENT(m);
```

pFlowClust *Parallelized FlowClust*

Description

A parallelized call to flowClust via the snow package and framework. Not called by the user.

Usage

```
pFlowClust(flowData,cl, K = 1:15, B.init = 100, tol.init = 0.01, tol = 1e-05, B
```

Arguments

`flowData` The data object, must be a flowFrame, flowSet or list of flowFrames

`cl` The snow cluster object

`K` The number of clusters to try for each flowFrame. Can be a vector. This is what is parallelized across processors.

`B.init` See `flowClust` documentation

`tol.init` See `flowClust` documentation

`tol` See `flowClust` documentation

`B` See `flowClust` documentation

`randomStart` See `flowClust` documentation

`nu` See `flowClust` documentation

`nu.est` See `flowClust` documentation

`trans` See `flowClust` documentation

`varNames` See `flowClust` documentation

Details

Calls flowClust via the `clusterMap` method of the `snow` package. Parallelizes the computation of multiple components for a single flowFrame in a loop over multiple flowFrames. If the snow cluster is NULL, will make the call via mapply.

Value

Returns a list of lists of flowClust objects The outer list corresponds to the flowFrames passed into the method. The inner list corresponds to the `K` cluster solutions passed into the method, for each `flowFrame` (ie If the input is a list of two flowFrames, and K=1:10, then the result is a list of length 2. Each element of the list is itself a list of length 10. The kth element of the inner list is the flowClust k cluster solution.)

Author(s)

Greg Finak «`greg.finak@ircm.qc.ca`»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data (Submitted)

See Also

flowClust,**snow**

`pFlowMerge` *Parellel call to flowMerge*

Description

Calls the flowMerge methods to compute the merged solution from a flowClust object or set of objects in a parallelized manner using the `snow` framework.

Usage

```
pFlowMerge(flowData, cl, K = 1:15, B.init = 100, tol.init = 0.01, tol = 1e-05, B
```

Arguments

`flowData` The data to be fit. A list of `flowFrames`, a `flowSet` or a `flowFrame`

`cl` The `snow` cluster object. Can be NULL to call the non-parallel version of flowClust

`K` See `flowClust` documentation

`B.init` See `flowClust` documentation

`tol.init`	See `flowClust` documentation
`tol`	See `flowClust` documentation
`B`	See `flowClust` documentation
`randomStart`	See `flowClust` documentation
`nu`	See `flowClust` documentation
`nu.est`	See `flowClust` documentation
`trans`	See `flowClust` documentation
`varNames`	See `flowClust` documentation

Details

Makes a parallelized call to `flowClust`. Parses the results to extract the max BIC solution, merges clusters, finds the optimal k-cluster solution using the entropy and returns it. If `cl` is `NULL`, a non-parallel call is made to the flowClust function.

Value

A list of flowMerge objects. One per flowFrame passed into the method.

Warning

This function does not do any special memory management. A large data set will likely cause it to run out of memory and start swapping incessantly. If you have lots of data, it's best to feed it piecewise to pFlowClust.

Author(s)

Greg Finak «`greg.finak@ircm.qc.ca`»

References

Finak G, Bashasharti A, Brinkmann R, Gottardo R. Merging Mixture Model Components for Improved Cell Population Identification in High Throughput Flow Cytometry Data (Submitted)

See Also

pFlowClust,flowClust,merge,**snow**, fitPiecewiseLinreg

Examples

```
data(rituximab)
#Parallelized call below:
## Not run: cl<-makeSOCKcluster(rep("finakg@localhost",7))
## Not run: result<-pFlowMerge(rituximab,cl,varNames=c("FSC.H","SSC.H"))
## Not run: plot(result)
cl<-NULL;
result<-pFlowMerge(rituximab,cl=NULL,varNames=c("FSC.H","SSC.H"),K=1:8);
plot(result);
```

`plot-methods` *Methods for plotting flowMerge and flowObj classes*

Description

Plots all possible two-dimensional projections of the parameters in a `flowMerge` or `flowObj` object and does not require specification of the `flowFrame` since a pointer to the data is stored in the object. Informative axis names are used, rather than the usual FL1/FL2/FS/SS channel names. This funciton can take most of the usual additional arguments provided to `plot` for the `flowClust` package, although some, like the axis names and the `data` are fixed. In order for `flowMerge` objects to display outliers correctly with `plot` (following merging), the `updateU` method must be called on them first.

Methods

x = "flowMerge", y = "missing" `x` is a `flowMerge` object.

x = "flowObj", y = "missing" `x` is a `flowObj` object.

See Also

flowClust

Examples

```
data(rituximab)
data(RituximabFlowClustFit)
o<-flowObj(flowClust.res[[which.max(flowMerge:::BIC(flowClust.res))]],rituximab);
m<-merge(o);
i<-fitPiecewiseLinreg(m);
m<-m[[i]];
plot(m,pch=20,level=0.9);
```

`show-methods` *Describe a flowObj or flowMerge object*

Description

Accessors to describe a `flowObj` or `flowMerge` object.

Methods

object = "flowMerge" Describe a `flowMerge` object.

object = "flowObj" Describe a `flowObj` object.

`split-methods` *Split data in a flowMerge object by cluster*

Description

Split method defined for flowMerge objects. Pulls out the population based on cluster number.

Methods

\itemx = "missing", f = "flowMerge" Split a flowMerge object into its component clusters.

`summary-methods` *Summary methods for flowMerge*

Description

Summary method for `flowMerge` objects.

Methods

object = "flowMerge" Summarize a flowMerge object.

object = "flowObj" Summarize a flowObj object

`updateU` *Update uncertainties*

Description

Updates the uncertainties in a `flowMerge` ojbect after merging clusters. This function is now internal and no longer exported. It is called automatically within the cluster merging method.

Usage

```
updateU(object)
```

Arguments

`object`	An object of type `flowMerge`

Details

Updates the `u` slot of the `flowMerge` object following merging. The update is computation intensive, and so, is not automatically performed on each `flowMerge` object. Should only be done on objects used in further analysis.

Value

A `flowMerge` object with the `u` slot updated to reflect the new parameter values.

Author(s)

Greg Finak «greg.finak@ircm.qc.ca»

See Also

flowMerge-class,merge

Examples

```
data(rituximab)
data(RituximabFlowClustFit)
o<-flowObj(flowClust.res[[which.max(flowMerge:::BIC(flowClust.res))]],rituximab);
m<-merge(o);
i<-fitPiecewiseLinreg(m);
m<-m[[i]];
plot(m,pch=20,level=0.9);
```

Index