Package ‘immunoClust’

July 22, 2016

Type Package

Title immunoClust - Automated Pipeline for Population Detection in Flow Cytometry

Version 1.4.0

Date 2016-03-23

Author Till Soerensen <till-antoni.soerensen@charite.de>

Maintainer Till Soerensen <till-antoni.soerensen@charite.de>

Depends R(>= 3.2), methods, stats, graphics, grid, lattice, flowCore

Suggests BiocStyle

Description Model based clustering and meta-clustering of Flow Cytometry Data

Collate class.immunoClust.R plot.immunoClust.R splom.immunoClust.R data.R process.R cell.clustering.R meta.clustering.R meta.gating.R meta.export.R meta.plot.R clust.util.R transform.R

biocViews Clustering, FlowCytometry, CellBasedAssays

License Artistic-2.0

NeedsCompilation yes

R topics documented:

immunoClust-package 2
cell.ClustData 3
cell.EM 4
cell.FitModel 6
cell.hclust 7
cell.ME 8
cell.process 9
cell.removed 13
cell.SubClustering 14
dat.exp 16
dat.fcs 17

dat.meta 18
immunoClust-object 19
meta.clustering 20
meta.export 22
meta.exprs 24
meta.hclust 25
meta.ME 26
meta.normalize 27
meta.plot 29
meta.process 31
meta.SubClustering 33
plot.immunoClust 34
splom.immunoClust 36
trans.ApplyToData 37
trans.FitToData 39

Index **41**

`immunoClust-package` *immunoClust - Automated Pipeline for Population Detection in Flow Cytometry*

Description

Model based clustering and meta-custering routines for Flow Cytometry (FC) data.

The immunoClust-pipeline consits of two major procedures:

`cell.process`	Clustering of cell-events
`meta.process`	Meta-clustering of cell-clusters

Cell-events clustering is performed for each FC data sample separately. After this all cell-clustering results are collected in a vector and meta-clustering is performed to obtain the across samples populations.

Details

Package:	immunoClust
Type:	Package
Version:	1.0.0
Depends:	R(>= 2.13.0), methods, stats, graphics, grid, lattice, flowCore
Date:	2015-01-28
License:	Artistic-2.0

Author(s)

Till Sörensen <till-antoni.soerensen@charited.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

---

cell.ClustData — *Model Based Clustering of Data for a pre-defined Number of Clusters*

Description

Performs EM-iteration on cell events, where an initial event cluster membership is obtained by hierarchical clustering on a sample subset given a number of clusters.

Usage

```
cell.ClustData(data, K, parameters=NULL, expName="immunoClust Experiment",
                sample.seed=1, sample.number=1500, sample.standardize=TRUE,
                B=50, tol=1e-5, modelName="mvt")
```

Arguments

`data`	A numeric matrix, data frame of observations, or object of class flowFrame. Rows correspond to observations and columns correspond to measured parameters.
`K`	Given number of clusters for the final model.
`parameters`	A character vector specifying the parameters (columns) to be included in clustering. When it is left unspecified, all the parameters will be used.
`expName`	The name of the clustering experiment.
`sample.seed`	The seed integer for the random number generator.
`sample.number`	The maximum number of samples used for initial hierarchical clustering.
`sample.standardize`	A numeric indicating whether the samples for hierarchical clustering are standardized (mean=0, SD=1).
`B`	The maximum number of EM-iterations.
`tol`	The tolerance used to assess the convergence of the EM-algorithm.
`modelName`	Used mixture model; either `"mvt"` for a t-mixture model or `"mvn"` for a Gaussian Mixture model.

Details

Although this function provides the possiblity to cluster an abitrary set of observed data into a fixed number of clusters, this function is used in the immunoClust-pipeline only for the calculation of the initial model with one cluster.

Value

The fitted model cluster information in an object of class `immunoClust`.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

`immunoClust-object`, `cell.hclust`

Examples

```
data(dat.fcs)
res <- cell.ClustData(dat.fcs, parameters=c("FSC-A", "SSC-A"), 5)
summary(res)
```

`cell.EM` *immunoClust EMt-iteration on Cell-events given initial Model Parameters*

Description

Performs EMt-iteration on cell event observations giving initial model parameters and returns the fitted clusters information in an object of class `immunoClust`.

Usage

```
cell.EM(data, parameters=NULL, expName="immunoClust Experiment",
    history=NULL, state=NULL,
    K, w, m, s, B=50, tol=1e-5, bias=0.5, modelName="mvt")

cell.Estimation(data, parameters=NULL, expName="immunoClust Experiment",
    history=NULL, state=NULL,
    K, w, m, s, modelName="mvt")
```

Arguments

`data`	A numeric matrix, data frame of observations, or object of class flowFrame.
`parameters`	A character vector specifying the parameters (columns) to be included in clustering. When it is left unspecified, all the parameters will be used.
`expName`	The name of the clustering experiment.
`history`	experimental; unused so far.
`state`	experimental: unused so far.
`K`	The number of clusters.
`w`	The K-dimensional vector of the mixture proportions.
`m`	The KxP-dimensional matrix of the K estimated cluster means.
`s`	The $KxPxP$-dimensional matrix of the K estimated cluster covariance matrices.
`B`	The maximum number of EMt-iterations.
`tol`	The tolerance used to assess the convergence of the EMt-algorithms.
`bias`	The ICL-bias used in the EMt-algorithm.
`modelName`	Used mixture model; either `"mvt"` or `"mvn"` for a t- or Gaussian mixture model respectively.

Details

Whereas `cell.EM` performs a complete EMt-iteration, `cell.Estimate` only calculates the posterior probabilities and the Maximum-A-Posterior estimators of cluster membership for all events.

Value

The fitted clusters information in an object of class `immunoClust`.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

`cell.ME`, `cell.FitModel`

Examples

```
data(dat.fcs)
data(dat.exp)
## cell.clustering result for dat.fcs
r <- dat.exp[[1]]
summary(r)
## apply model parameter to all (unfiltered) events
dat.trans <- trans.ApplyToData(r, dat.fcs)
r2 <- cell.EM(dat.trans, parameters=r@parameters, K=r@K, w=r@w,m=r@mu,s=r@sigma)
summary(r2)
```

`cell.FitModel` *immunoClust EMt-iteration on Cell-events given initial Model Parameters*

Description

The function fits initial model parameters to specific observed cell event data. The function returns the cluster information of the fitted model in an object of class `immunoClust`.

Usage

```
cell.FitModel(x, data, B=50, tol=1e-5, bias=0.5, modelName="mvt" )

cell.Classify(x, data, modelName="mvt" )
```

Arguments

`x`	An immunoClust object with the initial model parameter ($parameters$, K, w, mu, $sigma$).
`data`	A numeric matrix, data frame of observations, or object of class flowFrame.
`B`	The maximum number of EMt-iterations.
`tol`	The tolerance used to assess the convergence of the EMt-algorithms.
`bias`	The ICL-bias used in the EMt-algorithm.
`modelName`	Used mixture model; either `"mvt"` or `"mvn"` for a t- or Gaussian mixture model respectively.

Details

These functions are wrapper of the functions `cell.EM` and `cell.Estimation`, when model cluster parameters are combined in an object of class `immunoClust` and are used in the iterative cell event clustering process `cell.process` of *immuno*Clust and are not intended to be called directly.

Value

The fitted model cluster information in an object of class `immunoClust`.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

cell.process, cell.EM, cell.Estimation

Examples

```
data(dat.fcs)
data(dat.exp)
r1 <- dat.exp[[1]]
dat.trans <- trans.ApplyToData(r1, dat.fcs)
r2 <- cell.FitModel(r1, dat.trans)
```

---

cell.hclust — *Hierarchical Model Based Clustering of Cell-events in the immunoClust-pipeline*

Description

Performs model based agglomerative clustering on cell event observations with weights. It is used in the interative cell event clustering approach of *immuno*Clust to obtain an initial cluster membership for the EM(t)-iteration.

Usage

```
cell.hclust(data, weights=NULL)
```

Arguments

data	The numeric $N \times P$-dimensional data matrix to cluster. Each row contains a P-dimensional overservation vector.
weights	The N-dimensional vector of optional weights to be applied for the overservations.

Details

This function is used internally in cell.TestSubCluster procedure of **immunoClust**.

Value

A numeric $(N-1) \times 2$-dimensional matrix which gives the minimum index for observations in each of the two clusters merged at the *i*th step in each row.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

cell.TestSubCluster, cell.process

Examples

```
data(dat.fcs)
inc <- sample(1:nrow(dat.fcs), 50)
result <- cell.hclust(exprs(dat.fcs)[inc,])
```

---

cell.ME — *immunoClust EM-iteration on Cell-events given initial Cluster Membership Assignment*

Description

Performs an EM-iteration on cell event observations given an initial cluster membership for the cell events and returns the fitted cluster information in an object of class immunoClust.

Usage

```
cell.ME(data, parameters=NULL, expName="immunoClust Experiment",
        history=NULL, state=NULL, label, B=50, tol=1e-5, modelName="mvt")
```

Arguments

data	A numeric matrix, data frame of observations, or object of class flowFrame.
parameters	A character vector specifying the parameters (columns) to be included in clustering. When it is left unspecified, all the parameters will be used.
expName	The name of the clustering experiment.
history	experimental; unused so far.
state	experimental: unused so far.

`label`	The N-dimensional vector containing the initial cluster membership. A label-number of 0 for an event indicates that this event is not initially assigned to a cluster.
`B`	The maximum number of EMt-iterations.
`tol`	The tolerance used to assess the convergence of the EMt-algorithms.
`modelName`	Used mixture model; either `"mvt"` or `"mvn"` for a t- or Gaussian mixture model respectively.

Value

The fitted clusters information in an object of class `immunoClust`.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

`cell.EM`

Examples

```
data(dat.fcs)
data(dat.exp)
## cell.clustering result for dat.fcs
r1 <- dat.exp[[1]]
summary(r1)
## apply model parameter to all (unfiltered) events
dat.trans <- trans.ApplyToData(r1, dat.fcs)
r2 <- cell.ME(dat.trans, parameters=r1@parameters, label=r1@label)
summary(r2)
```

`cell.process` *Clustering of Cell-events in the immunoClust-pipeline*

Description

This function performs iterative model based clustering on cell-event data. It takes the observed cell-event data as major input and returns an object of class `immunoClust`, which contains the fitted mixture model parameter and cluster membership information. The additional arguments control the routines for data preprocessing, major loop and EMt-iteration, the model refinement routine and transformation estimation.

Usage

```
cell.process(fcs, parameters=NULL,
    apply.compensation=FALSE, classify.all=FALSE,
    N=NULL, min.count=10, max.count=10, min=NULL, max=NULL,
    I.buildup=6, I.final=4, I.trans=I.buildup,
    modelName="mvt", tol=1e-5, bias=0.3,
    sub.tol= 1e-4, sub.bias=bias, sub.thres=bias, sub.samples=1500,
    sub.extract=0.8, sub.weights=1, sub.standardize=TRUE,
    trans.estimate=TRUE, trans.minclust=10,
    trans.a=0.01, trans.b=0.0, trans.parameters=NULL)

cell.MajorIterationLoop(dat, x=NULL, parameters=NULL,
    I.buildup=6, I.final=4,
    modelName="mvt", tol=1e-5, bias=0.3,
    sub.bias=bias, sub.thres=0.0, sub.tol=1e-4, sub.samples=1500,
    sub.extract=0.8, sub.weights=1, sub.EM="MEt", sub.standardize=TRUE, seed=1)

cell.MajorIterationTrans(fcs, x=NULL, parameters=NULL,
    I.buildup=6, I.final=4, I.trans=I.buildup,
    modelName="mvt", tol=1e-5, bias=0.3,
    sub.bias=bias, sub.thres=0.0, sub.tol=1e-4, sub.samples=1500,
    sub.extract=0.8, sub.weights=1, sub.EM="MEt", sub.standardize=TRUE, seed=1,
    trans.minclust=5, trans.a=0.01, trans.decade=-1, trans.scale=1.0,
    trans.proc="vsHtransAw")

cell.InitialModel(dat, parameters=NULL, trans.a = 0.01, trans.b = 0.0,
    trans.decade=-1, trans.scale=1.0)

cell.classifyAll(fcs, x, apply.compensation=FALSE)
```

Arguments

`fcs` An object of class flowFrame. Rows correspond to observations and columns correspond to measured parameters.

`dat` A numeric matrix, data frame of observations, or object of class flowFrame. Rows correspond to observations and columns correspond to measured parameters.

`x` An object of class `immunoClust`. Used as inital model int the major iteration loop. When left unspecified the simplest model containing 1 cluster is used as initial model.

Arguments for data pre and post processing:

`parameters` A character vector specifying the parameters (columns) to be included in clustering. When it is left unspecified, all the parameters will be used.

`apply.compensation` A numeric indicator whether the compensation matrix in the flowFrame should be applied.

`classify.all` A numeric indicator whether the removed over- and underexposed observations should also be classified after the clustering process.

`N` Maximum number of observations used for clustering. When unspecified or higher than the number of observations (i.e. rows) in dat, all observations are used for clustering, otherwise only the first `N` observations.

`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 value of each parameter will be used. To suppress filtering, it is set to -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 limit set for data filtering. Note that it is a vector of length equal to the number of parameters (columns), implying that a different value can be set for each parameter.

`max` The upper limit set for data filtering. Interpretation is similar to that of `min`.

Arguments for the major loop and EMt-iteration:

`I.buildup` The number of major iterations, where the number of used observations is doubled successively.

`I.final` The number of major iterations with all observations.

`I.trans` The number of iterations where transformation estimation is applied.

`modelName` Used mixture model; either `"mvt"` for a t-mixture model or `"mvn"` for a Gaussian Mixture model.

`tol` The tolerance used to assess the convergence of the major EM(t)-algorithms of all observations.

`bias` The ICL-bias used in the major EMt-algorithms of all observations.

Arguments for model refinement (sub-clustering):

`sub.tol` The tolerance used to assess the convergence of the EM-algorithms in the sub-clustering.

`sub.bias` The ICL-bias used in the sub-clustering EMt-algorithms, in general the same as the ICL-bias.

`sub.thres` Defines the threshold, below which an ICL-increase is meaningless. The threshold is given as the multiple (or fraction) of the costs for a single cluster.

`sub.samples` The number of samples used for initial hierarchical clustering.

`sub.extract` The threshold used for cluster data extraction.

`sub.weights` Power of weights applied to hierarchical clustering, where the used weights are the probabilities of cluster membership.

`sub.EM` Used EM-algorithm; either `"MEt"` for EMt-iteration or `"ME"` for EM-iteration without test step.

`sub.standardize` A numeric indicating whether the samples for hierarchical clustering are standardized (mean=0, SD=1).

`seed` The seed integer for the random number generator.

Arguments for transformation optimization:

`trans.estimate` A numeric indicator whether transformation estimation should be applied.

`trans.minclust` The minimum number of clusters required to start transformation estimation.

`trans.a` A numeric vector, giving the (initial) scaling a for the asinh-transformation $h(y) = asin(a \cdot y + b)$. A scaling factor of $a = 0$ indicates that a parameter is not transformed.

`trans.b` A numeric vector, giving the (initial) translation b for the asinh-transformation.

`trans.parameters` A character vector, specifying the parameters (columns) to be applied for transformation. When it is left unspecified, the parameters to be transformed are obtained by the `PxDISPLAY` information of the flowFrame description parameters. All parameters with LOG display values are transformed.

`trans.decade` A numeric scale value for the theorectical maximum of transformed observation value. If below 0, no scaling of the trasnformed values is applied, which is the default in the *immuno*Clust-pipeline.

`trans.scale` A numeric scaling factor for the linear (i.e. not transformed) parameters. By default the linear parameters (normally the scatter parameters) are not scaled.

`trans.proc` An experimental switch for alternative procedures; should be "vsHtransAw".

Details

The `cell.process` function does data preprocessing and calls the major iteration loop either with or without integrated transformation optimization. When transformation optimization is applied the transformation parameters give the initial transformation otherwise they define the fixed transformation.

The major iteration loop with included transformation optimization relies on `flowFrames` structure from the `flowCore`-package for the storage of the observed data.

The `cell.InitialModel` builds up an initial `immunoClust-object` with one cluster and the given transformation parameters.

The `cell.classifyAll` calculates the cluster membership for the removed cell events. The assigment of the cluster membership is critical for over- and underexposed obsevervations and the interpretaion is problematic.

Value

The fitted model information in an object of class `immunoClust`.

Note

a) The data preprocessing arguments (`min.count`, `max.count`, `min` and `max`) for removing over- and underexposed observations are adopted from `flowCust-package` with the same meaning.

b) The `sub.thres` value is given in here in relation to the single cluster costs $\frac{1}{2} \cdot P \cdot (P+1) \cdot log(N)$. An absolute increase of the log-likelihood above is reported as reasonable from the literature. From our experience a higher value is required for this increase in FC data. For the ICL-bias and the

sub.thres identical values were chosen. For the CyTOF dataset this value had been adjusted to 0.05 since the absolute increase of the log-likelihood became to high due to the high number of parameters.

c) The `sub.extract` value controls the smooth data extraction for a cluster. A higher value includes more events for a cluster in the sub-clustering routine.

d) The default value of `trans.a=0.01` for the initial transformation is optimized for Fluorescence Cytometry. For CyTOF data the initial scaling value was `trans.a=1.0`.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

`immunoClust-object`, `plot`, `splom`, `cell.FitModel`, `cell.SubClustering`, `trans.FitToData`

Examples

```
data(dat.fcs)
res <- cell.process(dat.fcs)
summary(res)
```

`cell.removed` *Brief Information of removed Cell-events by immunoClust Cell-event Clustering*

Description

Gives information about the amount of overexposed cell-event observetion in a FCS-file.

Usage

```
removed.above(fcs, parameters=NULL, N=NULL, max.count=10, max=NULL)
removed.below(fcs, parameters=NULL, N=NULL, min.count=10, min=NULL)
```

Arguments

`fcs`	An object of class flowFrame. Rows correspond to observations and columns correspond to measured parameters.
`parameters`	A character vector specifying the parameters (columns) to be included in clustering. When it is left unspecified, all the parameters will be used.
`N`	Maximum number of observations used for clustering. When unspecified or higher than the number of observations (i.e. rows) in dat, all observations are used for clustering, otherwise only the first `N` observations.
`max.count`	An integer specifying the threshold count for filtering data points from above. The default is 10, meaning that if 10 or more data points are larger than or equal to `max`, they will be excluded from the analysis. If `max` is `NULL`, then the maximum value of each parameter will be used. To suppress filtering, it is set to -1.
`max`	The upper limit set for data filtering. Note that it is a vector of length equal to the number of parameters (columns), implying that a different value can be set for each parameter.
`min.count`	analoguous to `max.count`.
`min`	analoguous to `min`.

Value

A table with two rows containing the number of events above `max` in each parameter and above in only this parameter. The two last columns give the sum and percentage of all events above `max` in any parameter.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

Examples

```
data(dat.fcs)
removed.above(dat.fcs)
```

`cell.SubClustering` *immunoClust Model Refinement Step in iterative Cell-events Clustering*

Description

These function tests each cell-cluster of a model for refining it into more sub-clusters and returns the refined model parameter in an object of class `immunoClust`.

Usage

```
cell.SubClustering( x, dat, B=50, tol=1e-5, thres=0.1, bias=0.5,
                    sample.weights=1, sample.EM="MEt",
                    sample.number=1500, sample.standardize=TRUE,
                    extract.thres=0.8, modelName="mvt")

cell.TestSubCluster(x, y, t, cluster, J=8, B=500, tol=1e-5, bias=0.5,
                    sample.EM="MEt", sample.df=5, sample.number=1500,
                    sample.standardize=TRUE, modelName="mvt")
```

Arguments

`x`	An immunoClust object with the initial model parameter (K, w, mu, $sigma$).
`dat`	A numeric matrix, data frame of observations, or object of class flowFrame.
`B`	The maximum number of EM(t)-iterations in Sub-Clustering.
`tol`	The tolerance used to assess the convergence of the EM(t)-algorithms in Sub-Clustering.
`thres`	Defines the threshold, below which an ICL-increase is meaningless. The threshold is given as the multiple (or fraction) of the costs for a single cluster.
`bias`	The ICL-bias used in the EMt-algorithm.
`sample.weights`	Power of weights applied to hierarchical clustering, where the used weights are the probabilities of cluster membership.
`sample.EM`	Used EM-algorithm; either `"MEt"` for EMt-iteration or `"ME"` for EM-iteration without test step.
`sample.number`	The number of samples used for initial hierarchical clustering.
`sample.standardize`	A numeric indicating whether the samples for hierarchical clustering are standardized (mean=0, SD=1).
`extract.thres`	The threshold used for cluster data extraction.
`modelName`	Used mixture model; either `mvt` for a t-mixture model or `mvn` for a Gaussian Mixture model.
`y`	A numeric matrix of the observations beloning to the particular cluster.
`t`	A numeric vector with the probability weights for the observations belonining to the particular cluster.
`cluster`	An integer index of the particular cluster
`J`	The number of sub-models to be builded and tested for a particular cluster.
`sample.df`	Degree of freedom for the t-distibutions in a t-mixture model. Has to be 5 in immunoClust.

Details

These function are used internally by the cell-clustering procedures of `cell.process` in *immuno*Clust and are not intended to be used directly.

Value

The cluster parameters of the refined model in an object of class immunoClust.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

cell.process, cell.hclust

Examples

```
data(dat.fcs)
data(dat.exp)
dat.trans <- trans.ApplyToData(dat.exp[[1]], dat.fcs)
#need to re-calculate the cluster membership probabilities
# not stored in dat.exp
r1 <- cell.Classify(dat.exp[[1]], dat.trans)
summary(r1)
r2 <- cell.SubClustering(r1, dat.trans)
summary(r2)
```

---

dat.exp — *immunoClust Meta-clustering Sample*

Description

A vector of immunoClust-objects with cell.process clustering results of five samples.

Usage

```
data("dat.exp")
```

Details

Cell-event clustering was performed on reduced (10.000 events) sample data of the dataset of *immuno*Clust, MACS-depleted populations datasets 2010. URL http://flowrepository.org/id/FR-FCM-ZZWB.

Value

A vector of 5 `immnoClust-objects` for the cell clustering results of 5 FC samples.

`[[1]]` CD19 MACS-depleted cells

`[[2]]` CD15 MACS-depleted cells

`[[3]]` CD14 MACS-depleted cells

`[[4]]` CD4 MACS-depleted cells

`[[4]]` CD3 MACS-depleted cells

Source

http://flowrepository.org/id/FR-FCM-ZZWB

Examples

```
data(dat.exp)

## process meta clustering
meta <- meta.process(dat.exp, meta.bias=0.6)

## extract event counts in the 5 samples for all meta clusters
res <- meta.numEvents(meta)
```

---

dat.fcs *immunoClust Cell-clustering Sample*

Description

`flowFrame` data sample with 10.000 events in 7 parameters.

Usage

```
data(dat.fcs)
```

Details

This FCS sample is a reduced (10.000 events) dataset in flowFrame format of the first sample in the dataset of immunoClust, MACS-depleted populations datasets 2010. URL http://flowrepository.org/id/FR-FCM-ZZWB.

Value

A `flowCore` `flowFrame` with 10.000 observations on the following 7 parameters.

`FCS-A` Forward scatter

`SSC-A` Sideward scatter

`FITC-A` CD14

`PE-A` CD19

`APC-A` CD15

`APC-Cy7-A` CD4

`Pacific Blue-A` CD3

Source

http://flowrepository.org/id/FR-FCM-ZZWB

Examples

```
data(dat.fcs)

## process cell clustering
dat.res <- cell.process(dat.fcs)

## apply asinh-transformation
dat.fcs.transformed <- trans.ApplyToData(dat.res, dat.fcs)

## plot results
splom(dat.res, dat.fcs.transformed)
```

---

dat.meta *immunoClust Meta-clustering Results Sample*

Description

The Meta-clustering result of the `dat.exp` data set.

Usage

```
data("dat.meta")
```

Details

The Meta-clustering was performed with an ICL-bias of 0.4.

Value

A list-object containing the meta-clusering result. A detailed description is documented in the value section for the `meta.process` function.

Source

http://flowrepository.org/id/FR-FCM-ZZWB

Examples

```
data(dat.meta)

## extract event counts in the 5 samples for all meta clusters
res <- meta.numEvents(dat.meta)
```

---

immunoClust-object *immunoClust-Object*

Description

The `immunoClust` object contains the clustering results in the *immuno*Clust-pipeline as obtained by `cell.process` or `meta.process`.

Usage

```
## S4 method for signature 'immunoClust'
summary(object)
## S4 method for signature 'immunoClust'
show(object)
```

Arguments

`object`	An object of class `immunoClust` as returned by the `cell.process` or `meta.process` functions of the *immuno*Clust-pipeline.

Value

An object of class `immunoClust` has the following slots:

`expName`	The name of the clustering experiment.
`fcsName`	The path of the clustered FCS-file.
`parameters`	The parameters used for clustering.
`removed.below`	Number of observations removed from below.
`removed.above`	Number of observations removed from above.
`trans.a`	The P-dimensional vector of the scaling factors for the asinh-transformation of each used parameter. A sca
`trans.b`	The P-dimensional vector of the translations for the asinh-transformation of each used parameter.
`trans.decade`	experimental; should be -1.
`trans.scale`	experimental; should be 1.0.
`K`	The number of clusters.
`N`	The number of observations.
`P`	The number of used parameters.
`w`	The K-dimensional vector of the mixture proportions.

mu	The KxP-dimensional matrix of the K estimated cluster means.
sigma	The $KxPxP$-dimensional matrix of the K estimated cluster covariance matrices.
z	The KxN-dimensional matrix containing the posterior probabilities of cluster membership.
label	The N-dimensional vector containing the maximum a posteriori estimator for cluster membership.
logLike	A vector of length 3 containing the BIC, ICL and the classification likelihood without penalty of the fitted r
BIC	The Bayesian Information Criterion for the fitted mixture model.
ICL	The Integrate Classification Likelihood for the fitted model.
history	experimental; unused so far.
state	experimental; unused so far.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

cell.process, meta.process

Examples

```
data(dat.exp)
summary(dat.exp[[1]])
```

---

meta.clustering *Clustering of Cell-clusters in the immunoClust-pipeline*

Description

This function provides a direct access to the meta-clustering procedure. The method described and discussed in this manuscript is the EMt-classification (EM-method=20) with the number of events for each cluster as weights. It returns the fitted mixture model parameter in an object of class immunoClust.

Usage

```
meta.Clustering(P, N, K, W, M, S, I.iter=10, B=500, tol=1e-5,
                bias=0.25, alpha=0.5, EM.method=20,
                norm.method=0, norm.blur=2, norm.minG=10)
```

Arguments

P	The number of observed parameters for the cell event clusters.
N	The number of cell-clustering experiments.
K	The N-dimensional vector with the numbers of cell event clusters in each experiment. The total number of clusters is $totK = sum_{i=1}^{K} K_i$.
W	The $totK$-dimensional vector with weights of all clusters.
M	The $totKxP$-dimensional matrix of all cluster means.
S	The $totKxPxP$-dimensional matrix of all cluster covariance matrices.
I.iter	The maximum number of major iteration steps.
B	The $totKxPxP$-dimensional matrix of all cluster covariance matrices.
tol	The tolerance used to assess the convergence of the EM(t)-algorithms.
bias	The ICL-bias used in the EMt-iteration of the meta-clustering.
alpha	A value between 0 and 1 used to balance the bhattacharrya probabilities calculated with either the full covariance matrices or using only the diagonal elements of it. When working with uncompensated FC data very high correlations between parameters may be observed due to spill over. This leads to a very low bhattacharrya probability for two clusters even if they are located nearby. Using a mixture of the probabilities calculated with the complete covariance matrices and the variance information of each parameter avoids this problem. With a value of alpha=1, only the probabilities with complete covariance matrices are applied. A reasonable value for alpha is 0.5.
EM.method	0 = KL-minimization not weighted 1 = BC-maximization not weighted 10 = BC-maximization weighted 2 = EMt-classification not weighted 20 = EMt-classification weighted
norm.method	Normalization function; see `meta.Normalize` for details.
norm.blur	For the normalization step the A-posterior probabilites of the cell-clusters belonging to a meta.clusters a used. In order to capture narrow cell-clusters reasonable the co-variance of the cell-clusters is blured for the A-posterior probabilities in the normalization step.
norm.minG	Minimum number of obtained meta-clusters required to process the normalization step in the major iteration loop.

Details

This function is used internally by the meta-clustering procedure `meta.process` in *immuno*Clust.

Value

The fitted model information in an object of class `immunoClust`.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

immunoClust-object, meta.SubClustering, meta.process

Examples

```
data(dat.exp)
d <- meta.exprs(dat.exp)
res <- meta.Clustering(d$P, d$N, d$K, d$clsEvents, d$M, d$S)
```

---

meta.export *immunoClust Meta-clustering Results Export*

Description

Thess functions collect the output of the meta.process and extracts the event numbers, relative frequencies or mean fluorescence intensities for each meta-cluster and cell-clustering experiment in a numeric table.

Usage

```
meta.numEvents(meta, out.all=TRUE, out.removed=FALSE,  out.unclassified=TRUE)
meta.relEvents(meta, out.all=TRUE)

meta.parMFI(meta, par, out.all=TRUE)

meta.numClusters(meta, out.all=TRUE)

meta.freqTable(meta)

meta.relEvents2(meta, major=1:5, out.all=TRUE)

meta.relEvents3(meta, major=1:5, out.all=TRUE)

meta.majorEvents(meta, major=1:6, out.events=TRUE)
```

Arguments

meta	The list-object returned by the function meta.process.
par	An integer index to the specific parameter.
out.all	A numeric indicator whether the event numbers of all hierarchical gating levels are obtained or only the meta-clusters themselves.

`out.removed` A numeric indcator whether the number of removed events, which are not used for clustering are exported.

`out.unclassified` A numeric indicator whether the event numbers of the hierarchical gating levels or all meta-clusters are exported.

`out.events` Switch between cell event numbers and relative cell frequencies.

`major` A numeric array indication the major scatter regions which were used as total events.

Value

A numberic matrix with

numEvents the number of cell events

relEvents relative frequencies, i.e. the number of cell events per total meeasured events

parMFI mean fluorecence intensities in one parameter, i.e. the meta-cluster centers in asinh-tranformed scale

numClusters the number of cell clusters

freqTable relative frequencies with respect to all gating hierarchie levels

relEvents2 prelimary function; as relEvents but is restricted to the events in the given major scatter regions.

relEvents3 prelimary function; union of relEvents and relEvents2

majorEvents the number of cell events for the major regions only

in each meta-cluster (and gating hierarchy level) for each cell-clustering experiment.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (submitted).

See Also

`meta.process`

Examples

```
data(dat.exp)
meta <- meta.process(dat.exp)
tbl <- meta.numEvents(meta)
```

`meta.exprs` *Collecting Data of an immunoClust vector*

Description

The function takes a vector of `immunoClust`-object obtained by the `cell.process` function and extracts ths information into a `list` object.

Usage

```
meta.exprs(exp, sub=c())
```

Arguments

`exp`	The vector of `immunoClust` object with the cell clustering results.
`sub`	A integer array indicating the parameter subset to be collected.

Value

A `list` object with the following slots:

`P`	The number of observed parameters for the cell event clusters.
`N`	The number of cell-clustering samples.
`K`	The N-dimensional vector with the numbers of cell event clusters in each experiment. The total number of clusters is $totK = sum_{i=1}^{N} K_i$.
`W`	The $totK$-dimensional vector with weights of all clusters.
`M`	The $totKxP$-dimensional matrix of all cluster means.
`S`	The $totKxPxP$-dimensional matrix of all cluster covariance matrices.
`expNames`	The N-dimensional vector with the experiment names of the cell clustering samples.
`expEvents`	The N-dimensional vector for the total number of events of the cell clustering samples.
`clsEvents`	The $totK$-dimensional vector for the event number of all clusters.
`desc`	The P-dimensional vector for the parameter description.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

immunoClust.

Examples

```
data(dat.exp)
d <- meta.exprs(dat.exp, sub=c(1,2))
```

---

meta.hclust *Hierarchical Meta-clustering of Cell-clusters in the immunoClust-pipeline*

Description

Performs agglomerative clustering on cell-clusters. It is used in the interative meta-clustering approach of *immuno*Clust to obtain an initial meta-cluster membership for the EM(t)-iteration.

Usage

```
meta.hclust(P, N, W, M, S)
```

Arguments

P	The number of parameters.
N	The number of clusters.
W	The N-dimensional vector with cluster weights, i.e. numbers of events in a cluster.
M	The $N \times P$-dimensional vector with cluster means.
S	The $N \times P \times P$-dimensional vector with cluster covariance matrices.

Details

This function is used internally in meta.TestSubCluster of **immunoClust**.

Value

A numeric $(N-1) \times 2$-dimensional matrix which gives the minimum index for observations in each of the two clusters merged at the *i*th step in each row.

Note

The merging distances need not to be monotonic increasing.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

meta.TestSubCluster, meta.process

Examples

```
data(dat.exp)
r <- dat.exp[[1]]
hcPairs <- meta.hclust(r@P, r@K, r@w, r@mu, t(apply(r@sigma,1,c)))
```

---

meta.ME *immunoClust EM(t)-iteration on Cell-clusters*

Description

Performs an EM(t)-iteration on cell-clusters given an initial meta-cluster membership for the cell-clusters and returns the fitted meta-clusters information in an object of class immunoClust.

Usage

```
meta.ME(P, N, K, W, M, S, label, B=100, tol=1e-5, method=20, bias=0.25,
    alpha=0.5, min.class=0)
```

Arguments

P	The number of observed parameters for the cell event clusters.
N	The number of cell-clustering experiments.
K	The N-dimensional vector with the numbers of cell event clusters in each experiment. The total number of clusters is $totK = sum_{i=1}^{K} K_i$.
W	The $totK$-dimensional vector with weights, i.e. number of events, of all clusters.
M	The $totKxP$-dimensional matrix of all cluster means.
S	The $totKxPxP$-dimensional matrix of all cluster covariance matrices.
label	The $totK$-dimension integer vector with the initial cell-cluster to meta-cluster membership.
B	The $totKxPxP$-dimensional matrix of all cluster covariance matrices.
tol	The tolerance used to assess the convergence of the EM(t)-algorithms.
method	0 = KL-minimization not weighted 1 = BC-maximization not weighted 10 = BC-maximization weighted 2 = EMt-classification not weighted 20 = EMt-classification weighted

bias	The ICL-bias used in the EMt-iteration of the meta-clustering.
alpha	A value between 0 and 1 used to balance the bhattacharrya probabilities calculated with either the full covariance matrices or using only the diagonal elements of it.
min.class	The minimum number of clusters for the final model.

Details

This function is used internally by the meta-clustering procedures `meta.process` and `meta.Clustering` in *immuno*Clust.

Value

The fitted meta-clusters information in an object of class `immunoClust`.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

`meta.process`, `meta.Clustering`

Examples

```
data(dat.exp)
d <- meta.exprs(dat.exp)
r <- meta.ME(d$P, d$N, d$K, d$clsEvents, d$M, d$S, label=rep(1,sum(d$K)))
```

`meta.normalize` *immunoClust normalization step with the meta.clustering process*

Description

Performs a normalization via linear regression of the cell-cluster samples to the meta-clustering model.

Usage

```
meta.Normalize(P, N, K, W, M, S, G, Z, method=3)
```

Arguments

`P`	The number of observed parameters for the cell event clusters.
`N`	The number of cell-clustering experiments.
`K`	The N-dimensional vector with the numbers of cell event clusters in each experiment. The total number of clusters is $totK = sum_{i=1}^{K} K_i$.
`W`	The $totK$-dimensional vector with weights, i.e. number of events, of all clusters.
`M`	The $totKxP$-dimensional matrix of all cluster means.
`S`	The $totKxPxP$-dimensional matrix of all cluster covariance matrices.
`G`	The number of meta-clusters.
`Z`	The $totKxG$-dimensional matrix with the A-Posterior probabilities for a cell-cluster belonging to a meta-cluster.
`method`	Alternative methods used for the normalization routine. Let Y denote the consensus meta-model build from all cell-event clusters of all experiments using the A-posterior Z and X the cell-event clusters in each experiment. 0 = no normalization 1 = $Y = a \times X$ 2 = $Y = a \times X + b$ 3 = $X = a \times Y$ 4 = $X = a \times Y + b$

Details

The regression used the cell-cluster and meta-cluster means weighted by the probabilities for a cell-cluster belonging to the meta-cluster. It builds a consensus meta-model from all cell-clusters using the A-posterior probabilities Z.

Value

Returns the normalized cell-clusters means and co-variance matrices in a list-object with the following slots:

P The number of observed parameters for the cell event clusters.

N The number of cell-clustering experiments.

K The N-dimensional vector with the numbers of cell event clusters in each experiment. The total number of clusters is $totK = sum_{i=1}^{K} K_i$.

W The $totK$-dimensional vector with weights, i.e. number of events, of all clusters.

M The $totKxP$-dimensional matrix of all cluster means.

S The $totKxPxP$-dimensional matrix of all cluster covariance matrices.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

See Also

meta.process, meta.Clustering

Examples

```
data(dat.meta)
dat <- dat.meta$dat.clusters
res <- dat.meta$res.clusters
dat.norm <- meta.Normalize(dat$P, dat$N, dat$K, dat$clsEvents,
    dat$M, dat$S, res@K, res@z)
```

---

meta.plot — *Ploting of immunoClust Meta-clustering Results*

Description

Thess functions collect plotting helper routinest of the results meta.process and relies on the prelimary gating which is also included with the meta.process function.

Usage

```
meta.plotClustersForScatter(meta, include, filter=0)

meta.plotClusters( meta, include=c() )

meta.plotScatter(meta)

meta.plotGate(meta, gating, pre, gates, pattern=c(),
png.size=1024, filter=0)

meta.plotGating(meta, pre, pattern=c(), png.size=1024, filter=0)

meta.plotExpResult(meta, exp, pattern=c(), png_pre, plot.ellipse=FALSE,
plot.class=FALSE, png.size=1024, filter=0, ellipse.quantile=0.95)

meta.plotExpClusters(meta, exp, include=c(), png_pre, png.size=1024,
plot.ellipse=FALSE, plot.class=FALSE, class.col=c(), filter=0,
ellipse.quantile=0.95, desc=NULL, N=NULL)
```

Arguments

meta	The list-object returned by the function meta.process.
exp	The vector of immunoClust-objects used for meta-clustering.
include	A numeric vector specifying which clusters will be shown on the plot. By default, all clusters are included.

`pattern`	A character array which give the decription path for the gate to be plotted.
`gating`	A list object containing the hierarchical gating information as generated within the meta-clustering process of immunoCust.
`pre`	File prefix for the output png image files. The name for the png-file is generated by concatinating `pre` with the description of the gate.
`png_pre`	Same as `pre` (see above).
`png.size`	Pixel width and height for the output the png image files.
`plot.ellipse`	A logical value indicating whether the cluster 90% percentil boundary is to be drawn or not.
`ellipse.quantile`	Quantile value for the clusters boundary ellipsis.
`plot.class`	Indicating whether the plotted cell-clusters are colored by the meta-clusters or individually.
`class.col`	Color(s) of the plotting cell-clusters. May specify a different color for each cluster.
`desc`	A character vector for the parameter description.
`N`	An integer for the maximum number of observations to be plotted. By default all observations are plotted.
`filter`	A numeric number to filter out meta-clusters which have less number of cell-clusters.
`gates`	An optional numeric matrix with the gate positions in each parameter.

Value

Plots the clustering results on an appropriatei plotting device or png-file.

meta.plotClustersForScatter Uses the scatter-clustering results of the meta object.

meta.plotClusters Short-cut for plotting the meta-clustering results.

meta.plotScatter Short-cut for plotting the scatter-clustering result.

meta.plotGate Plots the meta-clustering results in one hierachy level and iterates through the sub-levels of the gate.

meta.plotGating Plots the meta-clustering gating results in each hierarchy level stored in png-files in the directory given by `pre`.

meta.plotExpResults Plots the the cell-events belonginig to meta-clusters within a gate for each sample in a separate file. The gate is specified by `pattern`.

meta.plotExpClusters Plots the cell-events belonging the meta-clusters in `include` for each sample used for meta-clustering in a separate file.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

See Also

`meta.process`

Examples

```
data(dat.meta)
meta.plotClusters(dat.meta)
```

`meta.process` *Meta-clustering of Cell-clusters in the immunoClust-pipeline*

Description

This function performs iterative model based clustering on the clusters obtained by `cell.process` of several samples. Its input is a vector of the `immunoClust-objects` of the samples.

The function also performs in a secondary step an ordering of the meta-clusters according to their distribution in the scatter parameter and an automated gating process. These procedures are preliminary and not part of the presented algorithms of the reference.

Usage

```
meta.process(exp, dat.subset=c(), meta.iter=10, tol=1e-05, meta.bias=0.2,
            meta.alpha=.5, norm.method=0, norm.blur=2,
            scatter.subset=c(1,2), scatter.bias=0.25,
            scatter.prior=6)
```

Arguments

`exp`	A vector of `list` objects, each `list` contains the cell-clustering result of a sample in the `res` field. Addition fields are `name` and `fsc` containing the cell-sample name and fcs-filename, which are used for data output and plot routines.
`dat.subset`	A numeric vector defining the used observed parameters for the meta-clustering. If unset, all parameters in the cell-clustering results are used.
`meta.iter`	The number of major iterations.
`tol`	The tolerance used to assess the convergence of the EM(t)-algorithms.
`meta.bias`	The ICL-bias used in the EMt-iteration of the meta-clustering.
`meta.alpha`	A value between 0 and 1 used to balance the bhattacharrya probabilities calculated with either the full covariance matrices or using only the diagonal elements of it. When working with uncompensated FC data, very high correlations between parameters may be observed due to spill over. This leads to a very low bhattacharrya probability for two clusters even if they are located nearby. Using a mixture of the probabilities calculated with the complete covariance matrices and the variance information of each parameter avoids this problem. With a value of alpha=1, only the probabilities with complete covariance matrices are applied. A reasonable value for alpha is 0.5.
`norm.method`	A numeric selector for the normalization step to be performed during the major iteration.
`norm.blur`	The bluring constant by which the cell-clusters co-variance matrices are increased within the normalization step.

scatter.subset	A numeric vector, giving the indices for the scatter parameter. If the scatter.subset is empty, scatter clustering was not performed.
scatter.bias	The ICL-bias used in EMt-iteration of scatter-clustering.
scatter.prior	experimental; gives the number of initial scatter regions for scatter clustering.

Value

The function returns a list-object with the following components:

dat.clusters	A dat list-object of the cell event clusters used for meta-clustering.
res.clusters	The immunoClust-object of the fitted meta-clustering mixture model.
dat.scatter	A dat list-object of the scatter parameters for the cell event clusters used for scatter clustering.
res.scatter	The immunoClust-object of the fitted scatter-clustering mixture model.
gating	A list-object containing the hierarchical gating-tree.

The components of the dat list-objects are:

P	The number of parameters for the cell event clusters.
N	The number of cell-clustering experiments.
K	The N-dimensional vector with the numbers of cell event clusters in each experiment. The total number of clust
W	The $totK$-dimensional vector with the mixture proportions of all clusters.
M	The $totKxP$-dimensional matrix of all cluster means.
S	The $totKxPxP$-dimensional matrix of all cluster covariance matrices.
expNames	The N-dimensional character vector with the cell-clustering experiment names.
expEvents	The N-dimensional vector with the numbers of events in each cell-clustering experiment.
clsEvents	The $totK$-dimensional vector with the number of events in each cluster.
desc	The P-dimensional character vector with the parameter description.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

immunoClust-object, meta.Clustering, meta.export, cell.process

Examples

```
data(dat.exp)
meta <- meta.process(dat.exp)
summary(meta$res.clusters)
```

```
tbl <- meta.numEvents(meta)
```

`meta.SubClustering` *immunoClust Model Refinement Step in iterative Meta-clustering*

Description

These function tests each meta-cluster of a model for refining it into more sub-clusters and returns the refined cluster memberships in an integer array.s

Usage

```
meta.SubClustering(P, N, W, M, S, label, tol=1e-5, bias=0.25, alpha=1.0,
    EM.method=20)

meta.TestSubCluster(P, N, W, M, S, J=8, B=500, tol=1e-5, bias=0.5, alpha=1.0,
    EM.method=2, HC.samples=2000)
```

Arguments

`P`	The number of parameters.
`N`	The number of clusters.
`W`	The N-dimensional vector with cluster weights, i.e. numbers of events in a cluster.
`M`	The NxP-dimensional vector with cluster means.
`S`	The $NxPxP$-dimensional vector with the cluster covariance matrices.
`label`	The N-dimension integer vector with the cell-cluster to meta-cluster membership.
`tol`	The tolerance used to assess the convergence of the EM(t)-algorithms in Sub-Clustering.
`bias`	he ICL-bias used in the EMt-algorithm.
`alpha`	A value between 0 and 1 used to balance the bhattacharrya probabilities calculated with either the full covariance matrices or using only the diagonal elements of it.
`J`	The number of sub-models to be builded and tested for a particular cluster.
`B`	The maximum number of EM(t)-iterations in Sub-Clustering.
`EM.method`	0 = KL-minimization not weighted 1 = BC-maximization not weighted 10 = BC-maximization weighted 2 = EMt-classification not weighted 20 = EMt-classification weighted
`HC.samples`	The number of samples used for initial hierarchical clustering.

Details

These function are used internally by the meta-clustering procedures `meta.process` and `meta.Clustering` in *immuno*Clust and are not intended to be used directly.

Value

An integer array of length N containing the cell-clusters meta-cluster memberships of the refined model.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

`meta.process`, `meta.Clustering`, `meta.hclust`

Examples

```
data(dat.exp)
d <- meta.exprs(dat.exp)
label <- rep(1,sum(d$K))
label <- meta.SubClustering(d$P, sum(d$K), d$clsEvents, d$M, d$S, label=label)
## Not run:
r1 <- meta.ME(d$P, d$N, d$K, d$clsEvents, d

## End(Not run)
```

---

plot.immunoClust *Scatterplot of immunoClust Clustering Results*

Description

This method generates scatterplot revealing the cluster assignment.

Usage

```
## S4 method for signature 'immunoClust'
plot(x, data, subset=c(1,2), ellipse=T,
show.rm=F, include=1:(x@K), main=NULL,
col=include+1, pch=".", cex=0.6,
col.rm=1, pch.rm=1, cex.rm=0.6, ecol=col, elty=1,
npoints=501, add=F, ...)
```

Arguments

x	An object of class immunoClust as return by cell.process.
data	A matrix, data frame of observations, or object of class flowFrame. This is the object of observations on which cell.process was performed or the matrix of cell-cluster centers for the meta.process.
subset	A numeric vector of length two indicating which two parameters are selected for the scatterplot. Alternatively, a character vector containing the names of the two parameters is allowed if x@parameters is not NULL.
ellipse	A logical value indicating whether the cluster 90% percentil boundary is to be drawn 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.
col	Color(s) of the plotting points. May specify a different color for each cluster.
pch	Plotting character(s) of the plotting points. May specify a different character for each cluster.
cex	Size of the plotting characters. 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.
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.

Value

Plots the clustering assignment on an appropriatei plotting device.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

immunoClust-object

Examples

```
data(dat.fcs)
data(dat.exp)
dat.res <- dat.exp[[1]]
dat.trans <- trans.ApplyToData(dat.res, dat.fcs)
plot(dat.res, dat=dat.trans)
```

---

splom.immunoClust *Scatterplot Matrix of immunoClust Clustering Results*

Description

This method generates scatterplot matrix revealing the cluster assignment.

Usage

```
## S4 method for signature 'immunoClust,missing'
splom(x, data, include=1:(x@K), ...)

## S4 method for signature 'immunoClust,flowFrame'
splom(x, data, include=1:(x@K),
subset=1:length(attributes(x)$param), N=NULL,label=NULL, desc=NULL,
add.param=c(), ...)

## S4 method for signature 'immunoClust,matrix'
splom(x, data, include=1:(x@K),
subset=1:length(attributes(x)$param), N=NULL,label=NULL, desc=NULL, ...)

datSplom(label, data, subset=1:ncol(data), include=1:nrow(data), ...)
```

Arguments

x	An object of class immunoClust as return by cell.process or meta.process.
data	Missing, a matrix, or object of class flowFrame. This is the object of observations on which cell.process was performed.
include	A numeric vector specifying which clusters will be shown on the plot. By default, all clusters are included.
subset	A numeric vector indicating which parameters are selected for the scatterplot matrix.
N	An integer for the maximum number of observations to be plotted. By default all observations are plotted.

`label`	A integer vector for the cluster mebership of the observations. By default this is `x@label`.
`desc`	A character vector for the parameter description.
`add.param`	A list of additional parameters to plot, which are not used for clustering.
`...`	Further graphical parameters passed to the generic function `splom`.

Value

An object of class `trellis` as returned by the generic `splom` function of the `lattice`-package. The `print` method (called by default) will plot it on an appropriate plotting device.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

`immunoClust-object`

Examples

```
data(dat.fcs)
data(dat.exp)
# cell clustering results of dat.fcs
dat.res <- dat.exp[[1]]
dat.trans <- trans.ApplyToData(dat.res, dat.fcs)
splom(dat.res, data=dat.trans, N=1000)
```

`trans.ApplyToData` *immunoClust asinh-Transformation*

Description

Applies the transformation information of the `immunoClust` object to the raw observed FC dataset.

Usage

```
trans.ApplyToData(x, data, add.param=c(), max.decade=attr(x,"trans.decade"),
    lin.scale=attr(x,"trans.scale") )
```

Arguments

`x`	The `immunoClust` object containing the estimators for the transformation `trans.a` and `trans.b`.
`data`	The numeric matrix, data frame of observations, or object of class `flowFrame`.
`add.param`	A list of additional parameters in the `flowFrame`, which are not used for clustering but should be included in the final transformed resulting `flowFrame`.
`max.decade`	A numeric scale for the maximum transformed observation value; if missing or below 0, no scaling of the transformed values is apllied, which is the default in *immuno*Clust.
`lin.scale`	A numeric scaling factor for the linear, i.e. not transformed, parameters; if missing no scaling, i.e. $lin.scale = 1$, is applied, which is the default in *immuno*Clust.

Details

In *immuno*Clust an $asinh$-transformation $h(y) = asinh(a \cdot y + b)$ is applied to the fluorescence parameter in the observed data. The scatter parameter are assumed to be linear.

Value

A matrix or `flowFrame` with replaced transformed oberservation values.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

`immunoClust`, `trans.FitToData`, `cell.process`

Examples

```
data(dat.fcs)
data(dat.exp)
dat.trans <- trans.ApplyToData(dat.exp[[1]], dat.fcs)
## Not run:
plot(dat.exp[[1]], data=dat.trans)

## End(Not run)
```

`trans.FitToData` *immunoClust asinh-Transformation Optimization*

Description

Performs variance stabilization transformation estimation on the fluorescense parameters of the observed cell events. It is integrated in the interative cell event clustering approach of *immuno*Clust when transformation estimation should be applied.

Usage

```
trans.FitToData(x, data, B=10, tol=1e-5, certainty=0.3, proc="vsHtransAw")
```

Arguments

`x`	The `immunoClust` object of the fitted mixture model and initial estimators for the transformation.
`data`	The numeric matrix, data frame of observations, or object of class flowFrame.
`B`	The maximum number of BFG2 minimizer iterations.
`tol`	The tolerance used to assess the convergence for the BFG2 minimizer.
`certainty`	Minimum probability for cluster membership of an observation to be taken into account.
`proc`	An experimental switch for alternative procedures; should be "vsHtransAw".

Details

In *immuno*Clust an $asinh$-transformation $h(y) = asinh(a \cdot y + b)$ is applied for all fluorescence parameter in the observed data.

The transformation optimization `trans.FitToData` requires a fitted model of cluster information together with suitable initial transformation estimation in an `immunoClust` object. It fits the transformation based on the initial scaling values `trans.a` and translation values `trans.b` to the observed data. It returns the optimized transformation parameter in a $2 \times P$-dimensional matrix, first row for the scaling and second row for the translation values. A scaling value of $a = 0$ on input and output indicates, that a parameter should not be transformed.

The presented transformation optimization ("vsHtransAw") fits only the scaling value. An alternative procedure ("vsHtrans_w") fits both, the scaling and the translation value, but turns out to be less robust.

Value

Optimized transformation scaling and translation values in a $2 \times P$-dimensional matrix, first row for the scaling and second row for the translation values.

Author(s)

Till Sörensen <till-antoni.soerensen@charite.de>

References

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T. immunoClust - an automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry A* (accepted).

See Also

trans.ApplyToData, cell.process

Examples

```
data(dat.fcs)
data(dat.exp)
## in dat.exp the z-matrices of the immunoClust-object are removed
## so we have to re-calculate it first ...
dat.trans <- trans.ApplyToData(dat.exp[[1]], dat.fcs)
res <- cell.Classify(dat.exp[[1]], dat.trans)
## ... now the transformation parameter can be optimzed
trans.FitToData(res, dat.fcs)
```

Index

∗Topic **cluster**
cell.ClustData, 3
cell.EM, 4
cell.hclust, 7
cell.ME, 8
cell.process, 9
cell.SubClustering, 14
meta.clustering, 20
meta.hclust, 25
meta.ME, 26
meta.process, 31
meta.SubClustering, 33
∗Topic **datasets**
dat.exp, 16
dat.fcs, 17
dat.meta, 18
∗Topic **graphs**
plot.immunoClust, 34
splom.immunoClust, 36
∗Topic **package**
immunoClust-package, 2
∗Topic **print**
immunoClust-object, 19

cell.Classify *(*cell.FitModel*)*, 6
cell.classifyAll *(*cell.process*)*, 9
cell.ClustData, 3
cell.EM, 4, *7*, *9*
cell.Estimation, *7*
cell.Estimation *(*cell.EM*)*, 4
cell.FitModel, *5*, 6, *13*
cell.hclust, *4*, 7, *16*
cell.InitialModel *(*cell.process*)*, 9
cell.MajorIterationLoop *(*cell.process*)*, 9
cell.MajorIterationTrans *(*cell.process*)*, 9
cell.ME, *5*, 8
cell.process, *2*, *6–8*, 9, *15*, *16*, *19*, *20*, *31*, *32*, *35*, *36*, *38*, *40*
cell.removed, 13
cell.SubClustering, *13*, 14
cell.TestSubCluster, *7*, *8*
cell.TestSubCluster *(*cell.SubClustering*)*, 14

dat.exp, 16, *18*
dat.fcs, 17
dat.meta, 18
datSplom *(*splom.immunoClust*)*, 36

immunoClust, *4–6*, *8*, *9*, *12*, *14*, *16*, *21*, *25–27*, *35*, *36*, *38*, *39*
immunoClust *(*immunoClust-package*)*, 2
immunoClust-class *(*immunoClust-object*)*, 19
immunoClust-object, 19, *32*
immunoClust-package, 2
immunoClust.object *(*immunoClust-object*)*, 19
immunoClust.package *(*immunoClust-package*)*, 2

meta.Clustering, *27*, *29*, *32*, *34*
meta.Clustering *(*meta.clustering*)*, 20
meta.clustering, 20
meta.export, 22, *32*
meta.exprs, 24
meta.freqTable *(*meta.export*)*, 22
meta.hclust, 25, *34*
meta.majorEvents *(*meta.export*)*, 22
meta.ME, 26
meta.Normalize, *21*
meta.Normalize *(*meta.normalize*)*, 27
meta.normalize, 27
meta.numClusters *(*meta.export*)*, 22
meta.numEvents *(*meta.export*)*, 22
meta.parMFI *(*meta.export*)*, 22
meta.plot, 29
meta.plotClusters *(*meta.plot*)*, 29

meta.plotClustersForScatter
(meta.plot), 29
meta.plotExpClusters (meta.plot), 29
meta.plotExpResult (meta.plot), 29
meta.plotGate (meta.plot), 29
meta.plotGating (meta.plot), 29
meta.plotScatter (meta.plot), 29
meta.process, *2*, *18–23*, *26, 27*, *29, 30*, 31, *34*, *36*
meta.relEvents (meta.export), 22
meta.relEvents2 (meta.export), 22
meta.relEvents3 (meta.export), 22
meta.SubClustering, *22*, 33
meta.TestSubCluster, *25, 26*
meta.TestSubCluster
(meta.SubClustering), 33

plot, *13*
plot (plot.immunoClust), 34
plot,immunoClust,missing-method
(plot.immunoClust), 34
plot,immunoClust-method
(plot.immunoClust), 34
plot.immunoClust, 34

removed.above (cell.removed), 13
removed.below (cell.removed), 13

show (immunoClust-object), 19
show,immunoClust-method
(immunoClust-object), 19
show.immunoClust (immunoClust-object), 19
splom, *13*
splom (splom.immunoClust), 36
splom,immunoClust,flowFrame-method
(splom.immunoClust), 36
splom,immunoClust,matrix-method
(splom.immunoClust), 36
splom,immunoClust,missing-method
(splom.immunoClust), 36
splom.immunoClust, 36
summary (immunoClust-object), 19
summary,immunoClust-method
(immunoClust-object), 19
summary.immunoClust
(immunoClust-object), 19

trans.ApplyToData, 37, *40*
trans.FitToData, *13*, *38*, 39