Package ‘vsclust’

February 8, 2023

Type Package

Title Feature-based variance-sensitive quantitative clustering

Version 1.0.0

Date 2022-03-23

Description Feature-based variance-sensitive clustering of omics data. Optimizes cluster assignment by taking into account individual feature variance. Includes several modules for statistical testing, clustering and enrichment analysis.

License GPL-2

Imports matrixStats, limma, parallel, shiny, qvalue, grDevices, stats, MultiAssayExperiment, graphics

Suggests knitr, yaml, testthat (>= 3.0.0), rmarkdown, BiocStyle, clusterProfiler

LinkingTo Rcpp

biocViews Clustering, Annotation, PrincipalComponent, DifferentialExpression, Visualization, Proteomics, Metabolomics

VignetteBuilder knitr

Depends R (>= 4.2.0)

Config/testthat/edition 3

LazyData false

RoxygenNote 7.2.1

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

git_branch RELEASE_3_16

git_last_commit 53f16ce

git_last_commit_date 2022-11-01

Date/Publication 2023-02-07

Author Veit Schwaemmle [aut, cre]

Maintainer Veit Schwaemmle <veits@bmb.sdu.dk>

R topics documented:

`vsclust-package` *VSClust provides a powerful method to run variance-sensitive clustering*

Description

Clustering of high-dimensional quantitative data with data points that come with multiple measurements. In this clustering method, each feature is represented by a) its quantitative profile and b) its variance. Hence, the incertainty about a measurement enter in the determination of the most common patterns. This methods is both insensitive to noisy measurements and avoids finding clusters in homogeneously distributed data.

Details

The functions in this package comprise (i) methods to prepare the data for cluster analysis like statistical analysis (‘SignAnal‘and SignPairedAnal‘), PCA (‘PCAwithVar‘), (ii) direct application of the clustering algorithm on a (standardized) data matrix (‘vsclust_algorithm‘), (iii) for the further evaluation and visualization (such as ‘calcBHI‘ and ‘mfuzz.plot‘), and (iv) wrappers for the over workflow including statistical preparation (‘statWrapper‘), estimation of the cluster number (‘estimClustNum‘), running the clustering (‘runClustWrapper‘) and functional evaluation (‘runFuncEnrich‘).

Author(s)

Maintainer: Veit Schw\"ammle <veits@bmb.sdu.dk>

References

- Schw\"ammle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

- Schw\"ammle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

- Schw\"ammle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

`artificial_clusters` *Synthetic/artificial data comprising 5 clusters*

Description

10-dimensional data set with 500 simulating features measured over 5 replicates each, comprising a total of 50 samples. The first 250 features were modeled through normal distributions shifted in the 10-dimensional space to form 5 different clusters. The 2nd half of the features were modeled through a normal distribution around the origin and thus should be assigned to any cluster

Usage

```
artificial_clusters
```

Format

A data frame consisting of 500 features distributed over 5 clusters and being replicated 5 times each

Source

Protein Research Group, University of Southern Denmark, Odense `averageCond` *Calculate mean over replicates*

Description

Simple method to calculate the means for each feature across its replicates

Usage

```
averageCond(data, NumReps, NumCond)
```

Arguments

`data`	Matrix of data frame with numerical values. Columns corresponds to samples
`NumReps`	Number of replicates per experimental condition
`NumCond`	Number of different experimental conditions

Value

Matrix of data frame with averaged values over replicates for each conditions

Examples

```
data <- matrix(rnorm(1000), nrow=100)
av_data <- averageCond(data, NumCond=2, NumReps=5)
```

`calcBHI` *Calculate "biological homogeneity index"*

Description

This index is providing a number for the enriched GO terms and pathways to assess the biological content within a set of genes or proteins. The calculation is according to Datta, S. & Datta, S. Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. BMC bioinformatics 7, 397 (2006).

Usage

```
calcBHI(Accs, gos)
```

Arguments

`Accs`	list containing gene or protein IDs, such as UniProt accession names
`gos`	object from ClusterProfiler

Value

Biological Homogeneity Index

References

Datta, S. & Datta, S. Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. BMC bioinformatics 7, 397 (2006).

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
# Run enrichment analysis
data(gcSample, package="clusterProfiler")
xx <- clusterProfiler::compareCluster(gcSample, fun="enrichKEGG",
                     organism="hsa", pvalueCutoff=0.05)
# Generate random list from gcSample
rand_ids <- lapply(gcSample, function(x) sample(unlist(gcSample), 200))
calcBHI(rand_ids, xx)
```

ClustComp — *Function to run clustering with automatic fuzzifier settings (might become obsolete)*

Description

Run original fuzzy c-means and vsclust for a number of clusters and the given data set including data pre-processing and automatic setting of the data-dependent parameters like the lower limit of the fuzzifier.

Usage

```
ClustComp(
  dat,
  NSs = 10,
  NClust = NClust,
  Sds = Sds,
  cl = parallel::makePSOCKcluster(1)
)
```

Arguments

`dat`	a numeric data matrix
`NSs`	number of clusterings runs with different random seeds
`NClust`	Number of clusters
`Sds`	Standard deviation of features (either vector of the same length as features numbers in matrix or single value)
`cl`	object of class ‘cluster‘ or ‘SOCKcluster‘ to specify environment for parallelization

Value

List containing the objects

‘indices‘ containing minimum centroid distance and Xie-Beni index for both clustering methods

‘Bestcl‘ optimal vsclust results (variance-sensitive fcm clustering)

‘Bestcl2‘ optimal fuzzy c-means restults

‘m‘ vector of individual fuzzifer values per feature

‘withinerror‘ final optimization score for vsclust

‘withinerror2‘ final optimization score for fuzzy c-means clustering

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
#' # Generate some random data
data <- matrix(rnorm(seq_len(1000)), nrow=100)
# Run clustering
cl <- parallel::makePSOCKcluster(1, nnodes=1)
ClustCompOut <- ClustComp(data, cl=cl, NClust=6, Sds=1)
barplot(ClustCompOut$indices)
```

`cvalidate.xiebeni` *Xie Beni Index of clustering object*

Description

Calculate the Xie Beni index for validity of the cluster number in clustering results from running fuzzy c-means or vsclust original publication:

Usage

```
cvalidate.xiebeni(clres, m)
```

Arguments

`clres`	Output from clustering. Either fclust object or list containing the objects for ‘membership‘ and cluster ‘centers‘
`m`	Fuzzifier value

Value

Xie Beni index

References

Xie X.L., Beni G. (1991). A validity measure for fuzzy clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13, 841-847.

Examples

```
# Generate some random data
data <- matrix(rnorm(seq_len(1000)), nrow=100)
# Run clustering
clres <- vsclust_algorithm(data, centers=5, m=1.5)
# Calculate Xie-Beni index from results
cvalidate.xiebeni(clres, 1.5)
```

`determine_fuzz` *Determine individual fuzzifier values*

Description

This function calculated the values of the fuzzifier from a) the dimensions of the considered data set and b) from the individual feature standard deviations.

Usage

```
determine_fuzz(dims, NClust, Sds = 1)
```

Arguments

`dims`	vector of two integers containing the dimensions of the data matrix for the clustering
`NClust`	Number of cluster for running vsclust on (does no influence the calculation of ‘mm‘)
`Sds`	individual standard deviations, set to 1 if not available

Value

list of ‘m‘: individual fuzzifiers, ‘mm‘: standard fuzzifier for fcm clustering when not using vsclust algorithm

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15; 26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
# Generate some random data
data <- matrix(rnorm(seq_len(1000)), nrow=100)
# Estimate fuzzifiers
fuzz_out <- determine_fuzz(dim(data), 1)
# Run clustering
clres <- vsclust_algorithm(data, centers=5, m=fuzz_out$mm)
```

`estimClust.plot` *Plotting results from estimating the cluster number*

Description

This function visualizes the output from estimClustNumber, and there particularly the two validity indices Minimum Centroid Distance and Xie Beni Index.

Usage

```
estimClust.plot(ClustInd)
```

Arguments

`ClustInd`	Matrix with values from validity indices

Value

Multiple panels showing expression profiles of clustered features passing the minMem threshold

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. '2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
data("artificial_clusters")
dat <- averageCond(artificial_clusters, 5, 10)
dat <- scale(dat)
dat <- cbind(dat, 1)
ClustInd <- estimClustNum(dat, 6)
estimClust.plot(ClustInd)
```

---

estimClustNum *Wrapper for estimation of cluster number*

Description

This runs the clustering for different numbers of clusters, and estimates the most suitable numbers from applying the minimum centroid distance and the Xie Beni index. Multi-threading is used to shorten the computation times. Given the hierarchical structure of many data sets, the resulting numbers are suggestions. Inspection of the here plotted indices help to determine alternative cluster numbers, given by a strong decay of the minimum centroid distance and/or a low value of the Xie Beni index.

Usage

```
estimClustNum(dat, maxClust = 25, cores = 1)
```

Arguments

`dat`	matrix of features averaged over replicates. The last column contains their standard deviation
`maxClust`	Maximal number of cluster. The minimum is 3
`cores`	The number of threads to be used for parallelisation

Value

list with the items 'ClustInd': list of clustering objects for each number of clusters, 'p' plot object with plots for validity indices, 'numclust' optimal cluster number according to "minimum centroid distance"

Examples

```
data <- matrix(rnorm(1000), nrow=100)
estim_out <- estimClustNum(data, maxClust=10)
best_number <- max(estim_out[1])
```

---

mfuzz.plot *Plotting vsclust results*

Description

This function visualizes the clustered quantitative profiles in multiple figure panels. The parameters allow specifying the main items like axes labels and color maps. The code is adopted from the MFuzz package.

Usage

```
mfuzz.plot(
  dat,
  cl,
  mfrow = c(1, 1),
  colo,
  minMem = 0,
  timeLabels,
  filename = NA,
  xlab = "Time",
  ylab = "Expression changes"
)
```

Arguments

`dat`	a numeric data matrix containing the values used in the clustering
`cl`	clustering results from vsclust_algorithm or Bestcl object from clustComp function
`mfrow`	vector of two numbers for the number of rows and colums, figure panels are distributed in the plot
`colo`	color map to be used (can be missing)
`minMem`	filter for showing only features with a higher membership values than this value
`timeLabels`	alternative labels for different conditions
`filename`	for writing into pdf. Will write on screen when using NA
`xlab`	Label of x-axis
`ylab`	Label of y-axis

Value

Multiple panels showing expression profiles of clustered features passing the minMem threshold

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
#' # Generate some random data
data <- matrix(rnorm(seq_len(5000)), nrow=500)
# Run clustering
clres <- vsclust_algorithm(data, centers=2, m=1.5)
mfuzz.plot(data, clres,  mfrow=c(2,3), minMem=0.0)
```

---

optimalClustNum *Determine optimal cluster number from validity index*

Description

Calculated the optimal number from expected behavior of the indices. This would be a large decay for the Minimum Centroid Distance and a minimum for the Xie Beni index

Usage

```
optimalClustNum(ClustInd, index = "MinCentroidDist", method = "VSClust")
```

Arguments

`ClustInd`	Output from estimClustNum providing the calculated cluster validity indices
`index`	Either "MinCentroidDist" or "XieBeni"
`method`	Either "VSClust" or "FCM" for standard fuzzy c-means clustering

Value

optimal cluster number

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
  data("artificial_clusters")
  dat <- averageCond(artificial_clusters, 5, 10)
  dat <- scale(dat)
dat <- cbind(dat, 1)
ClustInd <- estimClustNum(dat, 6)
optimalClustNum
```

`pcaWithVar` *Visualize using principal component analysis (both loadings and scoring) including the variance from the replicates*

Description

The loading plot shows all features and their scaled variance. This provides an idea of the intrinsic noise in the data.

Usage

```
pcaWithVar(data, NumReps, NumCond, Sds = 1)
```

Arguments

`data`	Matrix of data frame with numerical values. Columns corresponds to samples
`NumReps`	Number of replicates per experimental condition
`NumCond`	Number of different experimental conditions
`Sds`	Standard deviation for each features. Usually using the one from LIMMA

Value

Loading and scoring plots that include feature variance

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
data <- matrix(rnorm(1000), nrow=100)
pcaWithVar(data, NumCond=2, NumReps=5, Sds=1)
```

`PrepareForVSClust` *Wrapper for statistical analysis*

Description

Prepare data for running vsclust clustering. This includes visualization running the functions for the principal component analysis and its visualization, statistical testing with LIMMA, as well as scaling and filtering of missing values

Usage

```
PrepareForVSClust(dat, NumReps, NumCond, isPaired = FALSE, isStat)
```

Arguments

`dat`	matrix or data frame of numerical data. Columns are samples. Replicates are grouped (i.e. A1, B1, C1, A2, B2, C2) when letters denote conditions and numbers the replicates. In case of ‘isStat=FALSE‘, you need a last column for the standard deviations
`NumReps`	Number replicates in the data
`NumCond`	Number of different experimental conditions. The total number of columns needs to be NumReps*NumCond
`isPaired`	Boolean for running paired or unpaired statistical tests
`isStat`	Boolean for whether to run statistical test or each column corresponds to a different experimental conditions. Then this function reads feature standard deviations from data frame from the last column

Value

list with the items ‘dat‘ (data matrix of features averaged over replicates and last column with their standard deviations), ‘qvals‘ FDRs from the statistical tests (each conditions versus the first), ‘StatFileOut‘ all of before for saving in file

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
data <- matrix(rnorm(2000), nrow=200)
stats <- PrepareForVSClust(data, 5, 2, isStat=TRUE)
```

PrepareSEForVSClust *Wrapper for statistical analysis for SummarizedExperiment object*

Description

Prepare data for running vsclust clustering. This includes visualization running the functions for the principal component analysis and its visualization, statistical testing with LIMMA, as well as scaling and filtering of missing values

Usage

```
PrepareSEForVSClust(
  se,
  assayname = 1,
  coldatname = NULL,
  isPaired = FALSE,
  isStat
)
```

Arguments

`se`	SummarizedExperiment object
`assayname`	Sample in SummarizedExperiment object
`coldatname`	Column in colData for extracting replicates
`isPaired`	Boolean for running paired or unpaired statistical tests
`isStat`	Boolean for whether to run statistical test or each column corresponds to a different experimental conditions. Then this function reads feature standard deviations from data frame from the last column

Value

list with the items ‘dat‘ (data matrix of features averaged over replicates and last column with their standard deviations), ‘qvals‘ FDRs from the statistical tests (each conditions versus the first), ‘StatFileOut‘ all of before for saving in file, ‘NumReps‘ number of replicates and ‘NumCond‘ number of different experimental conditions

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
data(miniACC, package="MultiAssayExperiment")

stats <- PrepareSEForVSClust(miniACC, coldatname="COC", isStat=TRUE)
```

---

protein_expressions *Data from a typical proteomics experiment*

Description

There are 12 samples coming from mouse fed with the four different diets, measured in three replicates each. Relative protein abundances were obtained using iTRAQ labelling. The given numbers are log2-transformed. Protein names as UniProt accession numbers are given as rownames.

Usage

```
protein_expressions
```

Format

A data frame consisting of 574 proteins measured in 12 samples:

HF.Rep.1 Mice fed with a high fat diet, replicate 1

HF.Rep.2 Mice fed with a high fat diet, replicate 2

HF.Rep.3 Mice fed with a high fat diet, replicate 3

TTA.Rep.1 Mice fed with a diet containing TTA (Tetradecylthioacetic Acid) high fat diet, replicate 1

TTA.Rep.2 Mice fed with a diet containing TTA (Tetradecylthioacetic Acid) high fat diet, replicate 2

TTA.Rep.3 Mice fed with a diet containing TTA (Tetradecylthioacetic Acid) high fat diet, replicate 3

FO.Rep.1 Mice fed with a fish oil diet, replicate 1

FO.Rep.2 Mice fed with a fish oil diet, replicate 2

FO.Rep.3 Mice fed with a fish oil diet, replicate 3

TTA.FO.Rep.1 Mice fed with a diet containing fish oil and TTA, replicate 1

TTA.FO.Rep.2 Mice fed with a diet containing fish oil and TTA, replicate 2

TTA.FO.Rep.3 Mice fed with a diet containing fish oil and TTA, replicate 3

Source

Protein Research Group, University of Southern Denmark, Odense

---

runClustWrapper *Wrapper for running cluster analysis*

Description

This function runs the clustering and visualizes the results.

Usage

```
runClustWrapper(dat, NClust, proteins = NULL, VSClust = TRUE, cores)
```

Arguments

`dat`	matrix or data frame with feature values for different conditions
`NClust`	Number of cluster for running the clustering
`proteins`	vector with additional feature information (default is NULL) to be added to the results
`VSClust`	boolean. TRUE for running the variance-sensitive clustering. Otherwise, the function will call standard fuzzy c-means clustering
`cores`	Number of threads for the parallelization

Value

list with the items ‘dat‘(the original data), ‘Bestcl‘ clustering results (same as from vsclust_algorithm), ‘p‘ (plot object with mfuzz plots), ‘outFileClust‘(suitable matrix with complete information) , ‘ClustInd‘ (information about being member of any cluster, feature needs on membership values > 0.5)

Examples

```
data(iris)
data <- cbind(iris[,seq_len(4)],1)
clust_out <- runClustWrapper(data, NClust=3, cores=1)
clust_out$p
```

---

runVSClustApp *Run VSClust as Shiny app*

Description

You will get the full functionality of the VSClust workflow with multiple visualizations and downloads

Usage

```
runVSClustApp()
```

Value

The shiny app should open in a browser or in RStudio.

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
runVSClustApp()
```

SignAnalysis *Unpaired statistical testing*

Description

Statistical testing and variance estimation in multi-dimensional data set. given by a matrix. This functions runs LIMMA paired tests and calculated the shrunken variance estimates.

Usage

```
SignAnalysis(Data, NumCond, NumReps)
```

Arguments

Data	a numeric data matrix with columns as samples. Different experimental conditions are grouped together in their replicates. The number of samples per group needs to be identical
NumCond	Number of different experimental conditions
NumReps	Number of replicates per experimental condition

Value

List containing the objects

‘pvalues‘ p-values before correction for multiple testing

‘qvalues‘ false discovery rates after correction for multiple testing (‘qvalue‘ method from ‘qvalue‘ library)

‘Sds‘ General standard deviation within replicates after using shrinkage by LIMMA

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
#' # Generate some random data
data <- matrix(rnorm(seq_len(1000)), nrow=100)
# Run statistical testing
stat_out <- SignAnalysis(data, 2, 5)
# Histogram of qvalues (no significant events)
hist(stat_out$qvalues, 50, xlab="q-values")
```

SignAnalysisPaired *Paired statistical testing*

Description

Statistical testing and variance estimation in multi-dimensional data set. given by a matrix. This functions runs LIMMA paired tests and calculated the shrunken variance estimates.

Usage

```
SignAnalysisPaired(Data, NumCond, NumReps)
```

Arguments

`Data`	a numeric data matrix with columns as samples. Different experimental conditions are grouped together in their replicates. The number of samples per group needs to be identical
`NumCond`	Number of different experimental conditions
`NumReps`	Number of replicates per experimental condition

Value

List containing the objects

‘qvalues‘ false discovery rates after correction for multiple testing (‘qvalue‘ method from ‘qvalue‘ library)

‘Sds‘ General standard deviation within replicates after using shrinkage (eBayes) by LIMMA

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
#' # Generate some random data with three different experimental conditions
data <- matrix(rnorm(seq_len(1500)), nrow=100)
# Run statistical testing
stat_out <- SignAnalysisPaired(data, 3, 5)
# Histogram of qvalues comparing the second to the first condition
hist(stat_out$qvalues[,1], 50, xlab="q-values")
```

SwitchOrder *arrange cluster member numbers from largest to smallest*

Description

arrange cluster member numbers from largest to smallest

Usage

```
SwitchOrder(Bestcl, NClust)
```

Arguments

Bestcl	fclust object
NClust	Number of clusters

Value

fclust object with reorder clusters

Examples

```
# Generate some random data
data <- matrix(rnorm(seq_len(1000)), nrow=100)
# Run clustering
clres <- vsclust_algorithm(data, centers=5, m=1.5)
clres <- SwitchOrder(clres, 5)
```

vsclust_algorithm *Run the vsclust clustering algorithm*

Description

This function calls the c++ implementation of the vsclust algorithm, being an extension of fuzzy c-means clustering with additional variance control and capability to run on data with missing values

Usage

```
vsclust_algorithm(
  x,
  centers,
  iterMax = 100,
  verbose = FALSE,
  dist = "euclidean",
  m = 2,
  ratePar = NULL,
```

```
  weights = 1,
  control = list()
)
```

Arguments

x	a numeric data matrix
centers	Either numeric for number of clusters or numeric matrix with center coordinates
iterMax	Numeric for maximum number of iterations
verbose	Verbose information
dist	Distance to use for the calculation. We prefer "euclidean" (default)
m	Fuzzifier value: numeric or vector of length equal to number of rows of x
ratePar	(experimental) numeric value for punishing missing values
weights	numeric or vector of length equal to number of rows of x
control	list with arguments to vsclust algorithms (now only cutoff for relative tolerance: reltol)

Value

list with details about clustering having the objects ‘centers‘ (positions of centroids), ‘size‘ (feature number per cluster), ‘cluster‘ (nearest cluster of each feature), ‘membership‘ matrix of membership values, ‘iter‘ (number of carried out iterations), ‘withinerror‘ (final error from optimization), ‘call‘(call of function)

References

Schwaemmle V, Jensen ON. VSClust: feature-based variance-sensitive clustering of omics data. Bioinformatics. 2018 Sep 1;34(17):2965-2972. doi: 10.1093/bioinformatics/bty224. PMID: 29635359.

Schwaemmle V, Hagensen CE. A Tutorial for Variance-Sensitive Clustering and the Quantitative Analysis of Protein Complexes. Methods Mol Biol. 2021;2228:433-451. doi: 10.1007/978-1-0716-1024-4_30. PMID: 33950508.

Schwaemmle V, Jensen ON. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics. 2010 Nov 15;26(22):2841-8. doi: 10.1093/bioinformatics/btq534. Epub 2010 Sep 29. PMID: 20880957.

Examples

```
#' # Generate some random data
data <- matrix(rnorm(seq_len(1000)), nrow=100)
# Run clustering
clres <- vsclust_algorithm(data, centers=5, m=1.5)
head(clres$membership)
```

Index