Package 'IntClust'

July 30, 2018

Type Package

Title Integration of Multiple Data Sets with Clustering Techniques

Version 0.1.0

Date 2018-07-19

Author Marijke Van Moerbeke

Maintainer Marijke Van Moerbeke <marijke.vanmoerbeke@uhasselt.be>

Description Several integrative data methods in which information of objects from different data sources can be combined are included in the IntClust package. As a single data source is limited in its point of view, this provides more insight and the opportunity to investigate how the variables are interconnected. Clustering techniques are to be applied to the combined information. For now, only agglomerative hierarchical clustering is implemented. Further, differential gene expression and pathway analysis can be conducted on the clusters. Plotting functions are available to visualize and compare results of the different methods.

License GPL-3

LazyData true

Imports ade4,a4Core, Biobase, cluster, plotrix, plyr, gplots, gridExtra, limma, gtools,e1071,pls,stats,utils,graphics,FactoMineR,analogue,lsa, SNFtool,grDevices,ggplot2,circlize,Rdpack,data.table,igraph

RdMacros Rdpack

Suggests MLP, biomaRt, org.Hs.eg.db, a4Base

RoxygenNote 6.0.1

NeedsCompilation no

Depends R (>= 2.10)

Repository CRAN

Date/Publication 2018-07-30 12:10:15 UTC

R topics documented:

`ABC.SingleInMultiple` *Single-source ABC clustering*

Description

The Aggregating Bundles of Clusters (ABC, (Amaratunga, Cabrera, and Kovtun 2008)) was originally developed for a single gene expression data. ABC is an iterative algorithm in which for each iteration a random sample of objects and features is taken of each data set. A clustering algorithm is run on each subset and an incidence matrix C is set up by dividing the resulting dendrogram in k clusters. After r iterations, all incidence matrices are summed and divided by number of times two objects were selected simultaneously. This similarity value is transformed into a dissimilarity measure expressing the number of times the objects are not clustered together when both are selected. The obtained matrix is used a input into a clustering algorithm.

Usage

```
ABC.SingleInMultiple(data, transpose = TRUE, distmeasure = "euclidean",
  weighting = FALSE, stat = "var", normalize = FALSE, method = NULL,
  gr = c(), bag = TRUE, numsim = 1000, numvar = 100, linkage = "ward",
  alpha = 0.625, NC = NULL, NC2 = NULL, mds = FALSE)
```

Arguments

data	A data matrix. It is assumed the rows are corresponding with the objects.
transpose	Logical, whether the data should be transposed to have the ABC orginal format of rows being the variables and columns the samples. Defaults to TRUE.
distmeasure	The distance measurs to be used for the data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to "euclidean".
weighting	Logical value indicating whether the rows should be weighted in the resampling.
stat	The statistic to be used in weighing the rows. Currently the Coefficient of Variation and Variance are allowed. The corresponding inputs for these should be, "cv" and "var". If the rows are to be weighed equally, any other string will do.
normalize	Logical. Indicates whether to normalize the distance matrices or not, default is FALSE. This is recommended if different distance types are used. More details on normalization in Normalization
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is NULL.
gr	A prespecified grouping of the samples to be used in calculating the F-statistic if stat="F".
bag	Logical, indicating whether the columns should be bagged in each iteration. Defaults to TRUE.
numsim	The number of iterations to be used in the ABC Algorithm. Default is 1000.
numvar	The number of featurus to be used at each iteration to calculate the temporary clusters in the ABC Algorithm.
linkage	Choice of inter group dissimilarity (character). Defaults to "ward".
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
NC	Expected number of clusters in the data; passed to Wards Method in each iteration. Default is NULL.
NC2	Expected number of clusters in the data; passed to Wards Method in the final calculation of the clusters. By default set to NULL such that NC2=NC. If NC2="syl", a silhouette will be used to determine the most likely number of clusters.
mds	Logical, indicating whether the dissimilarities calculated in the ABC Algorithm should be plotted using Multi Dimensional Scaling. Defaults to FALSE.

Value

The returned value is a list of two elements:

DistM	The resulting distance matrix matrix
Clust	The resulting clustering

The value has class 'Ensemble'.

References

Amaratunga D, Cabrera J and Kovtun V (2008). "Microarray learning with ABC." *Biostatistics*, **9**, pp. 128-136.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_ABC=ABC.SingleInMultiple(data=fingerprintMat,transpose=TRUE,distmeasure="tanimoto",
weighting=TRUE,stat="var", gr=c(),bag=TRUE, numsim=100,numvar=100,linkage="flexible",
alpha=0.625,NC=7, NC2=NULL, mds=FALSE)
```

ADC *Aggregated data clustering*

Description

Aggregated Data Clustering (ADC) is a direct clustering multi-source technique. ADC merges the columns of all data sets into a single large data set on which a final clustering is performed.

Usage

```
ADC(List, distmeasure = "tanimoto", normalize = FALSE, method = NULL,
  clust = "agnes", linkage = "flexible", alpha = 0.625)
```

Arguments

List	A list of data matrices of the same type. It is assumed the rows are corresponding with the objects.
distmeasure	Choice of metric for the dissimilarity matrix (character). Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to "tanimoto".
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to FALSE. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is NULL.

clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character). Defaults to "flexible".
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible".

Details

In order to perform aggregated data clustering, the ADC function was written. A list of data matrices of the same type (continuous or binary) is required as input which are combined into a single (larger) matrix. Hierarchical clustering is performed with the agnes function and the ward link on the resulting data matrix and an applicable distance measure is indicated by the user.

Value

The returned value is a list with the following three elements.

AllData	Fused data matrix of the data matrices
DistM	The distance matrix computed from the AllData element
Clust	The resulting clustering

The value has class "ADC". The Clust element will be of interest for further applications.

References

Fodeh J, Brandt C, Luong BT, Haddad A, Schultz M, Murphy T and Krauthammer M (2013). "Complementary Ensemble Clustering of Biomedical Data." *Journal of Biomedical Informatics*, **46**(3), pp. 436-443.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)
MCF7_ADC=ADC(List=L,distmeasure="tanimoto",normalize=FALSE,method=NULL,clust="agnes",
linkage="flexible",alpha=0.625)
```

ADEC *Aggregated data ensemble clustering*

Description

Aggregated Data Ensemble Clustering (ADEC) is a direct clustering multi-source technique. ADEC is an iterative procedure which starts with the merging of the data sets. In each iteration, a random sample of the features is selected and/or a resulting dendrogram is divided into k clusters for a range of values of k.

Usage

```
ADEC(List, distmeasure = "tanimoto", normalize = FALSE, method = NULL,
  t = 10, r = NULL, nrclusters = NULL, clust = "agnes",
  linkage = "flexible", alpha = 0.625)
```

Arguments

List	A list of data matrices of the same type. It is assumed the rows are corresponding with the objects.
distmeasure	Choice of metric for the dissimilarity matrix (character). Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to "tanimoto".
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to FALSE. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is NULL.
t	The number of iterations. Defaults to 10.
r	The number of features to take for the random sample. If NULL (default), all features are considered.
nrclusters	A sequence of numbers of clusters to cut the dendrogram in. If NULL (default), the function stops.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character). Defaults to "flexible".
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible".

Details

If r is specified and nrclusters is a fixed number, only a random sampling of the features will be performed for the t iterations (ADECa). If r is NULL and the nrclusters is a sequence, the clustering is performedon all features and the dendrogam is divided into clusters for the values of nrclusters (ADECb). If both r is specified and nrclusters is a sequence, the combination is performed (ADECc). After every iteration, either be random sampling, multiple divisions of the dendrogram or both, an incidence matrix is set up. All incidence matrices are summed and represent the distance matrix on which a final clustering is performed.

Value

The returned value is a list with the following three elements.

AllData	Fused data matrix of the data matrices
DistM	The resulting co-association matrix
Clust	The resulting clustering

The value has class 'ADEC'. The Clust element will be of interest for further applications.

References

Fodeh J, Brandt C, Luong BT, Haddad A, Schultz M, Murphy T and Krauthammer M (2013). "Complementary Ensemble Clustering of Biomedical Data." *Journal of Biomedical Informatics*, **46**(3), pp. 436-443.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)
MCF7_ADEC=ADEC(List=L,distmeasure="tanimoto",normalize=FALSE,method=NULL,t=100,
r=100,nrcluster=seq(1,10,1),clust="agnes",linkage="flexible",alpha=0.625)
```

BinFeaturesPlot_MultipleData

Visualization of characteristic binary features of multiple data sets

---

Description

A tool to visualize characteristic binary features of a set of objects in comparison with the remaining objects for multiple data sets. The result is a matrix with coloured cells. Columns represent objects and rows represent the specified features. A feature which is present is give a coloured cell while an absent feature is represented by a grey cell. The labels on the right indicate the names of the features while the labels on the bottom are the names of the objects.

Usage

```
BinFeaturesPlot_MultipleData(leadCpds, orderLab, features = list(),
  data = list(), validate = NULL, colorLab = NULL, nrclusters = NULL,
  cols = NULL, name = c("Data1", "Data2"), colors1 = c("gray90", "blue"),
  colors2 = c("gray90", "green"), margins = c(5.5, 3.5, 0.5, 5.5),
  cexB = 0.8, cexL = 0.8, cexR = 0.8, spaceNames = 0.2,
  plottype = "new", location = NULL)
```

Arguments

leadCpds	A character vector with the names of the objects in a first group, i.e., the group for which the specified features are characteristic. Default is NULL.
orderLab	A character vector with the order of the objects. Default is NULL.
features	A list with as elements character vectors with the names of the features to be visualized for each data set. Default is NULL.
data	A list with the different data sets. Default is NULL.
validate	Optional. A list with validation data sets. If a feature has a validation reference, these are added in a red colour. Default is NULL.
colorLab	Optional. A clustering object if the objects are to be coloured accoring to their clustering order. Default is NULL.

nrclusters	Optional. The number of clusters to divide the dendrogram of ColorLab. Default is NULL.
cols	Optional. A character vector with the colours of the different clusters. Default is NULL.
name	A character string with the names of the data sets. Default is c("Data1" ,"Data2") for two data sets.
colors1	A character vector with the colours to indicate the presence (first element) or the absence of the features for the objects in LeadCpds. Default is c('gray90','blue').
colors2	A character vector with the colours to indicate the presence (first element) or the absence of the features for the objects in the remaining objects. Default is c('gray90','green').
margins	A vector with the margings of the plot. Default is c(5.5,3.5,0.5,5.5).
cexB	The font size of the labels on the bottom: the object labels. Default is 0.80.
cexL	The font size of the labels on the left: the data labels. Default is 0.80.
cexR	The font size of the labels on the right: the feature labels. Default is 0.80.
spaceNames	A percentage of the height of the figure to be reserved for the names of the objects. Default is 0.20.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	Optional. If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

Comps=FindCluster(list(MCF7_F),nrclusters=10,select=c(1,8))

MCF7_Char=CharacteristicFeatures(List=NULL,Selection=Comps,binData=
list(fingerprintMat,targetMat),datanames=c("FP","TP"),nrclusters=NULL,
topC=NULL,sign=0.05,fusionsLog=TRUE,weightclust=TRUE,names=c("FP","TP"))

FeatFP=MCF7_Char$Selection$Characteristics$FP$TopFeat$Names[c(1:10)]
FeatTP=MCF7_Char$Selection$Characteristics$TP$TopFeat$Names[c(1:10)]

BinFeaturesPlot_MultipleData(leadCpds=Comps,orderLab=MCF7_Char$Selection$
objects$OrderedCpds,features=list(FeatFP,FeatTP),data=list(fingerprintMat,targetMat),
validate=NULL,colorLab=NULL,nrclusters=NULL,cols=NULL,name=c("FP","TP"),colors1=
c('gray90','blue'),colors2=c('gray90','green'),margins=c(5.5,3.5,0.5,5.5),cexB=0.80,
```

```
cexL=0.80,cexR=0.80,spaceNames=0.20,plottype="new",location=NULL)

## End(Not run)
```

`BinFeaturesPlot_SingleData` *Visualization of characteristic binary features of a single data set*

Description

A tool to visualize characteristic binary features of a set of objects in comparison with the remaining objects for a single data set. The result is a matrix with coloured cells. Columns represent objects and rows represent the specified features. A feature which is present is give a coloured cell while an absent feature is represented by a grey cell. The labels on the right indicate the names of the features while the labels on the bottom are the names of the objects.

Usage

```
BinFeaturesPlot_SingleData(leadCpds = c(), orderLab = c(), features = c(),
  data = NULL, colorLab = NULL, nrclusters = NULL, cols = NULL,
  name = c("Data"), colors1 = c("gray90", "blue"), colors2 = c("gray90",
  "green"), highlightFeat = NULL, margins = c(5.5, 3.5, 0.5, 5.5),
  plottype = "new", location = NULL)
```

Arguments

`leadCpds`	A character vector with the names of the objects in a first group, i.e., the group for which the specified features are characteristic. Default is NULL.
`orderLab`	A character vector with the order of the objects. Default is NULL.
`features`	A character vector with the names of the features to be visualized. Default is NULL.
`data`	The data matrix. Default is NULL.
`colorLab`	Optional. A clustering object if the objects are to be coloured accoring to their clustering order. Default is NULL.
`nrclusters`	Optional. The number of clusters to divide the dendrogram of ColorLab. Default is NULL.
`cols`	Optional. A character vector with the colours of the different clusters. Default is NULL.
`name`	A character string with the name of the data. Default is "Data".
`colors1`	A character vector with the colours to indicate the presence (first element) or the absence of the features for the objects in LeadCpds. Default is c('gray90','blue').
`colors2`	A character vector with the colours to indicate the presence (first element) or the absence of the features for the objects in the remaining objects. Default is c('gray90','green').

highlightFeat	Optional. A character vector with names of features to be highlighted. The names of the features are coloured purple. Default is NULL.
margins	A vector with the margings of the plot. Default is c(5.5,3.5,0.5,5.5).
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	Optional. If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Examples

```
## Not run:
data(fingerprintMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

Comps=FindCluster(list(MCF7_F),nrclusters=10,select=c(1,8))

MCF7_Char=CharacteristicFeatures(List=list(fingerprintMat),Selection=Comps,
binData=list(fingerprintMat),datanames=c("FP"),nrclusters=NULL,topC=NULL,
sign=0.05,fusionsLog=TRUE,weightclust=TRUE,names=c("FP"))Feat=MCF7_Char$
Selection$Characteristics$FP$TopFeat$Names[c(1:10)]

BinFeaturesPlot_SingleData(leadCpds=Comps,orderLab=MCF7_Char$Selection$
objects$OrderedCpds,features=Feat,data=fingerprintMat,colorLab=NULL,
nrclusters=NULL,cols=NULL,name=c("FP"),colors1=c('gray90','blue'),colors2=
c('gray90','green'),highlightFeat=NULL,margins=c(5.5,3.5,0.5,5.5),
plottype="new",location=NULL)

## End(Not run)
```

BoxPlotDistance *Box plots of one distance matrix categorized against another distance matrix.*

Description

Given two distance matrices, the function categorizes one distance matrix and produces a box plot from the other distance matrix against the created categories. The option is available to choose one of the plots or to have both plots. The function also works on outputs from ADEC and CEC functions which do not have distance matrices but incidence matrices.

Usage

```
BoxPlotDistance(Data1, Data2, type = c("data", "dist", "clusters"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), lab1, lab2, limits1 = NULL, limits2 = NULL,
  plot = 1, StopRange = FALSE, plottype = "new", location = NULL)
```

Arguments

Data1	The first data matrix, cluster outcome or distance matrix to be plotted.
Data2	The second data matrix, cluster outcome or distance matrix to be plotted.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
lab1	The label to plot for Data1.
lab2	The label to plot for Data2.
limits1	The limits for the categories of Data1.
limits2	The limits for the categories of Data2.
plot	The type of plots: 1 - Plot the values of Data1 versus the categories of Data2. 2 - Plot the values of Data2 versus the categories of Data1. 3 - Plot both types 1 and 2.
StopRange	Logical. Indicates whether the distance matrices with values not between zero and one should be standardized to have so. If FALSE the range normalization is performed. See Normalization. If TRUE, the distance matrices are not changed. This is recommended if different types of data are used such that these are comparable. Default is FALSE.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Value

One/multiple box plots.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)


BoxPlotDistance(MCF7_F,MCF7_T,type="cluster",lab1="FP",lab2="TP",limits1=c(0.3,0.7),
limits2=c(0.3,0.7),plot=1,StopRange=FALSE,plottype="new", location=NULL)

## End(Not run)
```

CEC *Complementary ensemble clustering*

Description

Complementary Ensemble Clustering (CEC) Complementary Ensemble Clustering (CEC, *Fodeh2013*) shows similarities with ADEC. However, instead of merging the data matrices, ensemble clustering is performedon each data matrix separately. The resulting incidence matrices for each data sets are combined in a weighted linear equation. The weighted incidence matrix is the input for the final clustering algorithm. Similarly as ADEC, there are versions depending of the specification of the number of features to sample and the number of clusters.

Usage

```
CEC(List, distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE,
  FALSE), method = c(NULL, NULL), t = 10, r = NULL, nrclusters = NULL,
  weight = NULL, clust = "agnes", linkage = c("flexible", "flexible"),
  alpha = 0.625, weightclust = 0.5)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
t	The number of iterations. Defaults to 10.

r	A vector with the number of features to take for the random sample for each element in List. If NULL (default), all features are considered.
nrclusters	A list with a sequence of numbers of clusters to cut the dendrogram in for each element in List. If NULL (default), the function stops.
weight	The weights for the weighted linear combination.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
weightclust	A weight for which the result will be put aside of the other results. This was done for comparative reason and easy access.

Details

If r is specified and nrclusters is a fixed number, only a random sampling of the features will be performed for the t iterations (CECa). If r is NULL and the nrclusters is a sequence, the clustering is performedon all features and the dendrogam is divided into clusters for the values of nrclusters (CECb). If both r is specified and nrclusters is a sequence, the combination is performed (CECc). After every iteration, either be random sampling, multiple divisions of the dendrogram or both, an incidence matrix is set up. All incidence matrices are summed and represent the distance matrix on which a final clustering is performed.

Value

The returned value is a list of four elements:

DistM	The resulting incidence matrix
Results	The hierarchical clustering result for each element in WeightedDist
Clust	The result for the weight specified in Clustweight

The value has class 'CEC'.

References

Fodeh J, Brandt C, Luong BT, Haddad A, Schultz M, Murphy T and Krauthammer M (2013). "Complementary Ensemble Clustering of Biomedical Data." *Journal of Biomedical Informatics*, **46**(3), pp. 436-443.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_CEC=CEC(List=L,distmeasure=c("tanimoto","tanimoto"),normalize=FALSE,method=NULL
,t=100, r=c(100,100), nrclusters=list(seq(2,10,1),seq(2,10,1)),clust="agnes",linkage=
c("flexible","flexible"),alpha=0.625,weightclust=0.5)
```

`CharacteristicFeatures` *Determining the characteristic features of a cluster*

Description

The function `CharacteristicFeatures` requires as input a list of one or multiple clustering results. It is capable of selecting the binary features which determine a cluster with the help of the fisher's exact test.

Usage

```
CharacteristicFeatures(List, Selection = NULL, binData = NULL,
  contData = NULL, datanames = NULL, nrclusters = NULL, sign = 0.05,
  topChar = NULL, fusionsLog = TRUE, weightclust = TRUE, names = NULL)
```

Arguments

`List`	A list of the clustering outputs to be compared. The first element of the list will be used as the reference in `ReorderToReference`.
`Selection`	If differential gene expression should be investigated for a specific selection of objects, this selection can be provided here. Selection can be of the type "character" (names of the objects) or "numeric" (the number of specific cluster). Default is NULL.
`binData`	A list of the binary feature data matrices. These will be evaluated with the fisher's extact test. Default is NULL.
`contData`	A list of continuous data sets of the objects. These will be evaluated with the t-test. Default is NULL.
`datanames`	A vector with the names of the data matrices. Default is NULL.
`nrclusters`	Optional. The number of clusters to cut the dendrogram in. The number of clusters should not be specified if the interest lies only in a specific selection of objects which is known by name. Otherwise, it is required. Default is NULL.
`sign`	The significance level to be handled. Default is 0.05.
`topChar`	Overrules sign. The number of features to display for each cluster. If not specified, only the significant genes are shown. Default is NULL.
`fusionsLog`	Logical. To be handed to `ReorderToReference`: indicator for the fusion of clusters. Default is TRUE
`weightclust`	Logical. To be handed to `ReorderToReference`: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
`names`	Optional. Names of the methods. Default is NULL.

Details

The function rearranges the clusters of the methods to a reference method such that a comparison is made easier. Given a list of methods, it calls upon `ReorderToReference` to rearrange the number of clusters according to the first element of the list which will be used as the reference.

Value

The returned value is a list with an element per method. Each element contains a list per cluster with the following elements:

`objects` A list with the elements LeadCpds (the objects of interest) and OrderedCpds (all objects in the order of the clustering result)

`Characteristics` A list with an element per defined binary data matrix in BinData and continuous data in ContData. Each element is again a list with the elements TopFeat (a table with information on the top features) and AllFeat (a table with information on all features)

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(geneMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_T ,MCF7_F)

MCF7_Char=CharacteristicFeatures(List=L,Selection=NULL,BinData=list(fingerprintMat,
targetMat),datanames=c("FP","TP"),nrclusters=7,topC=NULL,sign=0.05,fusionsLog=TRUE,
weightclust=TRUE,names=c("FP","TP"))

## End(Not run)
```

ChooseCluster *Interactive plot to determine DE Genes and DE features for a specific cluster*

Description

If desired, the function produced a dendrogram of a clustering results. One or multiple cluster can be indicated by a mouse click. From these clusters DE genes and characteristic features are determined. It is also possible to provide the objects of interest without producing the plot. Note, it is required to click on the dendrogram branches, not on the objects. #' @export ChooseCluster

Usage

```
ChooseCluster(Interactive = TRUE, leadCpds = NULL, clusterResult = NULL,
  colorLab = NULL, binData = NULL, contData = NULL, datanames = c("FP"),
  geneExpr = NULL, topChar = 20, topG = 20, sign = 0.05,
  nrclusters = NULL, cols = NULL, n = 1)
```

Arguments

Interactive	Logical. Whether an interactive plot should be made. Defaults to TRUE.
leadCpds	A list of the objects of the clusters of interest. If Interactive=TRUE, these are determined by the mouse-click and it defaults to NULL.
clusterResult	The output of one of the aggregated cluster functions, The clustering result of interest. Default is NULL.
colorLab	The clustering result the dendrogram should be colored after as in ClusterPlot. It is the output of one of the clustering functions.
binData	A list of the binary feature data matrices. These will be evaluated with the fisher's extact test. Default is NULL.
contData	A list of continuous data sets of the objects. These will be evaluated with the t-test. Default is NULL.
datanames	A vector with the names of the data matrices. Default is NULL.
geneExpr	A gene expression matrix, may also be an ExpressionSet. The rows should correspond with the genes. Default is NULL.
topChar	The number of top characteristics to return. If NULL, only the significant characteristics are saved. Default is NULL.
topG	The number of top genes to return. If NULL, only the significant genes are saved. Default is NULL.
sign	The significance level. Default is 0.05.
nrclusters	Optional. The number of clusters to cut the dendrogram in. If NULL, the dendrogram will be plotted without colors to discern the different clusters. Default is NULL.
cols	The colors to use in the dendrogram. Default is NULL.
n	The number of clusters one wants to identify by a mouse click. Default is 1.

Details

The DE genes are determined by testing for significance of the specified cluster versus all other objects combined. This is performed by the limma function. The binary features are evaluated with the fisher exact test while the continuous features are tested with the t-test. Multiplicity correction is included.

Value

The returned value is a list with one element per cluster of interest indicated by the prefix "Choice". This element is again a list with the following three elements:

objects	A list with the elements LeadCpds (the objects of interest) and OrderedCpds (all objects in the order of the clustering result)
Characteristics	The found (top) characteristics of the feature data
Genes	A list with the elements TopDE (a table with information on the top genes) and AllDE (a table with information on all genes)

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(geneMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

MCF7_Interactive=ChooseCluster(Interactive=TRUE,leadCpds=NULL,clusterResult=MCF7_T,
colorLab=MCF7_F,binData=list(fingerprintMat),datanames=c("FP"),geneExpr=geneMat,
topChar = 20, topG = 20,nrclusters=7,n=1)

## End(Not run)
```

Cluster *Single source clustering*

Description

The function Cluster performs clustering on a single source of information, i.e one data matrix. The option is available to compute the gap statistic to determine the optimal number of clusters.

Usage

```
Cluster(Data, type = c("data", "dist"), distmeasure = "tanimoto",
  normalize = FALSE, method = NULL, clust = "agnes",
  linkage = "flexible", alpha = 0.625, gap = TRUE, maxK = 15,
  StopRange = TRUE)
```

Arguments

Data	A matrix containing the data. It is assumed the rows are corresponding with the objects.
type	Type indicates whether the provided matrix in "Data" is either a data or a distance matrix obtained from the data. If type="dist" the calculation of the distance matrix is skipped. Type should be one of "data" or "dist".

distmeasure	Choice of metric for the dissimilarity matrix (character). Should be one of "tanimoto", "euclidean", "jaccard","hamming". Default is "tanimoto".
normalize	Logical. Indicates whether to normalize the distance matrices or not, default is FALSE. This is recommended if different distance types are used. More details on normalization in Normalization
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is NULL.
clust	Choice of clustering function (character). Defaults to "agnes". Note for now, the only option is to carry out agglomerative hierarchical clustering as it was implemented in the agnes function in the cluster package.
linkage	Choice of inter group dissimilarity (character). Defaults to "flexible".
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
gap	Logical. Whether the optimal number of clusters should be determined with the gap statistic. Default is TRUE.
maxK	The maximal number of clusters to investigate in the gap statistic. Default is 15.
StopRange	Logical. Indicates whether the distance matrices with values not between zero and one should be standardized to have so. #' If FALSE the range normalization is performed. See Normalization. If TRUE, the distance matrices are not changed. This is recommended if different types of data are used such that these are comparable. Default is TRUE.

Details

The gap statistic is determined by the criteria described by the cluster package: firstSEmax, globalSEmax, firstmax,globalmax, Tibs2001SEmax. The number of iterations is set to a default of 500. The implemented distances to be used for the dissimilarity matrix are jaccard, tanimoto and euclidean. The jaccard distances were computed with the dist.binary(...,method=1) function in the ade4 package and the euclidean ones with the daisy function in again the cluster package. The Tanimoto distances were implemented manually.

Value

The returned value is a list with two elements:

DistM	The distance matrix of the data matrix
Clust	The resulting clustering

If the gap option was indicated to be true, another 3 elements are joined to the list. Clust_gap contains the output from the function to compute the gap statistics and gapdata is a subset of this output. Both can be used to make plots to visualize the gap statistic. The final component is k which is a matrix containing the optimal number of clusters determined by each criterion mentioned earlier.

Examples

```
data(fingerprintMat)
data(targetMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",alpha=0.625,gap=FALSE,maxK=55
,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",alpha=0.625,gap=FALSE,maxK=55
,StopRange=FALSE)
```

ClusterCols *Matching clusters with colours*

Description

Internal function of ClusterPlot.

Usage

```
ClusterCols(x, Data, nrclusters = NULL, cols = NULL, colorComps = NULL)
```

Arguments

x	The leaf of a dendrogram.
Data	A clustering object.
nrclusters	The number of clusters to divide the dendrogram in. Default is NULL.
cols	A character vector with the colours to be used. Default is NULL.
colorComps	A character vector of a specific set of objects to be coloured. Default is NULL.

ClusteringAggregation *Clustering aggregation*

Description

The ClusteringAggregation includes the ensemble clustering methods Balls, Agglomerative (Aggl.) and Furthest which are graph-based consensus methods.

Usage

```
ClusteringAggregation(List, type = c("data", "dist", "clust"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), clust = "agnes", linkage = c("flexible",
  "flexible"), alpha = 0.625, nrclusters = c(7, 7), gap = FALSE,
  maxK = 15, agglMethod = c("Balls", "Aggl", "Furthest", "LocalSearch"),
  improve = TRUE, distThresh_B = 0.5, distThresh_A = 0.8)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
nrclusters	The number of clusters to divide each individual dendrogram in. Default is c(7,7) for two data sets.
gap	Logical. Whether the optimal number of clusters should be determined with the gap statistic. Default is FALSE.
maxK	The maximal number of clusters to investigate in the gap statistic. Default is 15.
agglMethod	The method to be performed: "Balls","Aggl","Furthest" or "LocalSearch".
improve	Logical. If TRUE, a local search is performed to improve the obtained results. Default is TRUE.
distThresh_B	A distance threshold for the Balls algoritme. Default is 0.5.
distThresh_A	A distance threshold for the Aggl. algoritme. Default is 0.8.

Details

(Gionis, Mannila, and Tsaparas 2007) propose heuristic algorithms in order to find a solution for the consensus problem. In a first step, a weighted graph is built from the objects with weights between two vertices determined by the fraction of clusterings that place the two vertices in different clusters. In a second step, an algorithm searches for the partition that minimizes the total number of disagreements with the given partitions. The Balls algorithm is an iterative process which finds a cluster for the consensus partition in each iteration. For each object i, all objects at a distance of at most 0.5 are collected and the average distance of this set to the ith object of interest is calculated. If the average distance is less or equal to a parameter $alpha$ the objects form a cluster; otherwise the object forms a singleton. The Agglomerative (Aggl.) algorithm starts by considering every object as a singleton cluster. Next, the two closest clusters are merged if the average distance between the clusters is less than 0.5. If there are no two clusters with an average distance smaller than 0.5, the algorithm stops and returns the created clusters as a solution. The Furthest algorithm

starts by placing all objects into a single cluster. In each iteration, the pair of objects that are the furthest apart are considered as new cluster centers. The remaining objects are appointed to the center that increases the cost of the partition the least and the new cost is computed. The cost is the sum of the all distances between the obtained partition and the partitions in the ensemble. The iteration continues until the cost of the new partition is higher than the previous partition.

Value

The returned value is a list of two elements:

`DistM`	A NULL object
`Clust`	The resulting clustering

The value has class 'Ensemble'.

References

Gionis A, Mannila H and Tsaparas P (2007). "Clustering aggregation." *ACM Transactions on Knowledge Discovery from Data*, **1**(1), pp. 4.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_Aggl=ClusteringAggregation(List=L,type="data",distmeasure=c("tanimoto","tanimoto"),
normalize=c(FALSE,FALSE),method=c(NULL,NULL),clust="agnes",linkage = c("flexible",
"flexible"),alpha=0.625,nrclusters=c(7,7),gap = FALSE, maxK = 15,agglMethod="Aggl",
improve=TRUE,distThresh_B=0.5,distThresh_A=0.8)
```

ClusterPlot *Colouring clusters in a dendrogram*

Description

Plot a dendrogram with leaves colored by a result of choice.

Usage

```
ClusterPlot(Data1, Data2 = NULL, nrclusters = NULL, cols = NULL,
  colorComps = NULL, hangdend = 0.02, plottype = "new", location = NULL,
  ...)
```

Arguments

Data1	The resulting clustering of a method which contains the dendrogram to be colored.
Data2	Optional. The resulting clustering of another method , i.e. the resulting clustering on which the colors should be based. Default is NULL.
nrclusters	Optional. The number of clusters to cut the dendrogram in. If not specified the dendrogram will be drawn without colours to discern the different clusters. Default is NULL.
cols	The colours for the clusters if nrclusters is specified. Default is NULL.
colorComps	If only a specific set of objects needs to be highlighted, this can be specified here. The objects should be given in a character vector. If specified, all other compound labels will be colored black. Default is NULL.
hangdend	A specification for the length of the brances of the dendrogram. Default is 0.02.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.
...	Other options which can be given to the plot function.

Value

A plot of the dendrogram of the first clustering result with colored leaves. If a second clustering result is given in Data2, the colors are based on this clustering result.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(Colors1)

MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

ClusterPlot(MCF7_T ,nrclusters=7,cols=Colors1,plottype="new",location=NULL,
main="Clustering on Target Predictions: Dendrogram",ylim=c(-0.1,1.8))

## End(Not run)
```

ColorPalette *Create a color palette to be used in the plots*

Description

In order to facilitate the visualization of the influence of the different methods on the clustering of the objects, colours can be used. The function `ColorPalette` is able to pick out as many colours as there are clusters. This is done with the help of the `ColorRampPalette` function of the grDevices package

Usage

```
ColorPalette(colors = c("red", "green"), ncols = 5)
```

Arguments

`colors`	A vector containing the colors of choice
`ncols`	The number of colors to be specified. If higher than the number of colors, it specifies colors in the region between the given colors.

Value

A vector containing the hex codes of the chosen colors.

Examples

```
Colors1<-ColorPalette(c("cadetblue2","chocolate","firebrick2",
"darkgoldenrod2", "darkgreen","blue2","darkorchid3","deeppink2"), ncols=8)
```

Colors1 *Colour examples*

Description

A vector of HEX codes for the colours used in the examples

Format

An object if class `"character"`.

Examples

```
data(Colors1)
```

ColorsNames *Function that annotates colors to their names*

Description

The ColorsNames function is used on the output of the ReorderToReference and matches the cluster numbers indicated by the cell with the names of the colors. This is necessary to produce the plot of the ComparePlot function and is therefore an internal function of this function but can also be applied separately.

Usage

```
ColorsNames(matrixColors, cols = NULL)
```

Arguments

matrixColors	The output of the ReorderToReference function.
cols	A character vector with the names of the colours to be used. Default is NULL.

Value

A matrix containing the hex code of the color that corresponds to each cell of the matrix to be colored. This function is called upon by the ComparePlot function.

Examples

```
data(fingerprintMat)
data(targetMat)
data(Colors1)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
names=c("FP","TP")

MatrixColors=ReorderToReference(List=L,nrclusters=7,fusionsLog=TRUE,weightclust=TRUE,
names=names)

Names=ColorsNames(matrixColors=MatrixColors,cols=Colors1)
```

`CompareInteractive` *Interactive comparison of clustering results for a specific cluster or method.*

Description

A visual comparison of all methods is handy to see which objects will always cluster together independent of the applied methods. The function `CompareInteractive` plots the comparison over the specified methods. A cluster or method can than be identified by clicking and is plotted separately against the single source or other specified methods.

Usage

```
CompareInteractive(ListM, ListS, nrclusters = NULL, cols = NULL,
  fusionsLogM = FALSE, fusionsLogS = FALSE, weightclustM = FALSE,
  weightclustS = FALSE, namesM = NULL, namesS = NULL, marginsM = c(2,
  2.5, 2, 2.5), marginsS = c(8, 2.5, 2, 2.5), Interactive = TRUE, n = 1)
```

Arguments

`ListM`	A list of the multiple source clustering or other methods to be compared and from which a cluster or method will be identified. The first element of the list will be used as the reference in `ReorderToReference`.
`ListS`	A list of the single source clustering or other methods the identified result will be compared to. The first element of the list will be used as the reference in `ReorderToReference`.
`nrclusters`	The number of clusters to cut the dendrogram in. Default is NULL.
`cols`	A character vector with the names of the colours. Default is NULL.
`fusionsLogM`	The fusionsLog parameter for the elements in ListM. To be handed to `ReorderToReference`. Default is FALSE.
`fusionsLogS`	The fusionslog parameter for the elements in ListS. To be handed to `ReorderToReference`. Default is FALSE.
`weightclustM`	The weightclust parameter for the elements in ListM. To be handed to `ReorderToReference`. Default is FALSE.
`weightclustS`	The weightclust parameter for the elements in ListS. To be handed to `ReorderToReference`. Default is FALSE.
`namesM`	Optional. Names of the multiple source clusterings to be used as labels for the columns. Default is NULL.
`namesS`	Optional. Names of the single source clusterings to be used as labels for the columns. Default is NULL.
`marginsM`	Optional. Margins to be used for the plot for the elements is ListM after the identification. Default is c(2,2.5,2,2.5).
`marginsS`	Optional. Margins to be used for the plot for the elements is ListS after the identification. Default is c(8,2.5,2,2.5).

Interactive	Optional. Do you want an interactive plot? Defaults to TRUE, if not the function provides the same as ComparePlot for the elements in ListM. Default is TRUE.
n	The number of methods/clusters you want to identify. Default is 1.

Value

The returned value is a plot of the comparison of the elements of ListM. On this plot multiple clusters and/or methods can be identified. Click on a cluster of a specific method to see how that cluster of that method compares to the elements in ListS. Click left next to a row to identify a all cluster of a specific method. A new plotting window will appear for every identification.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(Colors1)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(fingerprintMat,targetMat)

MCF7_W=WeightedClust(List=L,type="data",distmeasure=c("tanimoto","tanimoto"),
normalize=c(FALSE,FALSE),method=c(NULL,NULL),weight=seq(1,0,-0.1),weightclust=0.5,
clust="agnes",linkage="ward",StopRange=FALSE)

ListM=list(MCF7_W)
namesM=c(seq(1.0,0.0,-0.1))

ListS=list(MCF7_F,MCF7_T)
namesS=c("FP","TP")

CompareInteractive(ListM,ListS,nrclusters=7,cols=Colors1,fusionsLogM=FALSE,
fusionsLogS=FALSE,weightclustM=FALSE,weightclustS=TRUE,namesM,namesS,
marginsM=c(2,2.5,2,2.5),marginsS=c(8,2.5,2,2.5),Interactive=TRUE,n=1)

## End(Not run)
```

ComparePlot *Comparison of clustering results over multiple results*

Description

A visual comparison of all methods is handy to see which objects will always cluster together independent of the applied methods. To this aid the function ComparePlot has been written. The function relies on methods of the circlize package.

Usage

```
ComparePlot(List, nrclusters = NULL, cols = NULL, fusionsLog = FALSE,
  weightclust = FALSE, names = NULL, margins = c(8.1, 3.1, 3.1, 4.1),
  circle = FALSE, canvaslims = c(-1, 1, -1, 1), Highlight = NULL,
  plottype = "new", location = NULL)
```

Arguments

List	A list of the outputs from the methods to be compared. The first element of the list will be used as the reference in ReorderToReference.
nrclusters	The number of clusters to cut the dendrogram in. Default is NULL.
cols	A character vector with the colours to be used. Default is NULL.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE
weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
names	Optional. Names of the methods to be used as labels for the columns. Default is NULL.
margins	Optional. Margins to be used for the plot. Default is c(8.1,3.1,3.1,4.1).
circle	Logical. Whether the figure should be circular (TRUE) or a rectangle (FALSE). Default is FALSE.
canvaslims	The limits for the circular dendrogam. Default is c(-1.0,1.0,-1.0,1.0).
Highlight	Optional. A list of character vectors of objects to be highlighted. The names of the elements in the list are the names to appear on the figure. The median similarities of the objects in each list elemented is computed. Default is NULL.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	Optional. If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Details

This function makes use of the functions ReorderToReference and Colorsnames. Given a list with the outputs of several methods, the first step is to call upon ReorderToReference and to produce a matrix of which the columns are ordered according to the ordering of the objects of the first method in the list. Each cell represent the number of the cluster the object belongs to for a specific method indicated by the rows. The clusters are arranged in such a way that these correspond to that one cluster of the referenced method that they have the most in common with. The function color2D.matplot produces a plot of this matrix but needs a vector indicating the names of the colors to be used. This is where ColorsNames comes in. A vector of the color names of the output of the ReorderToReference is created and handed to color2D.matplot. It is optional to adjust

the margins of the plot and to give a vector with the names of the methods which will be used as labels for the rows in the plot. The labels for the columns are the names of the object in the order of clustering of the referenced method. Further, the similarity measures of the methods compared to the reference will be computed and shown on the right side of the plot.

Value

A plot which translates the matrix output of the function `ReorderToReference` in which the columns represent the objects in the ordering the referenced method and the rows the outputs of the given methods. Each cluster is given a distinct color. This way it can be easily observed which objects will cluster together. The labels on the right side of the plot are the similarity measures computed by `SimilarityMeasure`.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(Colors1)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
N=c("FP","TP")

#rectangular
ComparePlot(List=L,nrclusters=7,cols=Colors1,fusionsLog=TRUE,weightclust=TRUE,
names=N,margins=c(9.1,4.1,4.1,4.1),plottype="new",location=NULL)

#circle
Comps_I=c("fluphenazine","trifluoperazine","prochlorperazine","chlorpromazine")
Comps_II=c("butein","genistein","resveratrol")

ComparePlot(List=L,nrclusters=7,cols=c(Colors1), fusionsLog=TRUE,weightclust=FALSE,
names =N, margins = c(8.1, 3.1,3.1, 4.1),circle=TRUE,canvaslims=c(-1.1,1.1,-1.1,1.1),
Highlight=list("Comps I"=Comps_I,"Comps II"=Comps_II,"Cancer Treatments"=c("estradiol",
"fulvestrant")),plottype = "new")

## End(Not run)
```

`CompareSilCluster` *Compares medoid clustering results based on silhouette widths*

Description

The function `CompareSilCluster` compares the results of two medoid clusterings. The null hypothesis is that the clustering is identical. A test statistic is calcluated and a p-value obtained with bootstrapping. See "Details" for a more elaborate description.

Usage

```
CompareSilCluster(List, type = c("data", "dist"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), nrclusters = NULL, names = NULL, nboot = 100,
  plottype = "new", location = NULL)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
nrclusters	The number of clusters to cut the dendrogram in. This is necessary for the computation of the Jaccard coefficient. Default is NULL.
names	The labels to give to the elements in List. Default is NULL.
nboot	Number of bootstraps to be run. Default is 100.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Details

For the data or distance matrices in List, medoid clustering with nrclusters is set up by the pam function of the **cluster** and the silhouette widths are retrieved. These widths indicate how well an object fits in its current cluster. Values around one indicate an appropriate cluster while values around zero indicate that the object might as well lie in its neighbouring cluster. The silhouette widths are than regressed in function of the cluster membership of the objects. First the widths are modelled according to the cluster membership of object these were derived from. Next, these are modeled in function of the membership determined by the other object. The regression function is fit by the lm function and the r.squared value is retrieved. Ther.squared value indicates how much of the variance of the silhouette widths is explained by the membership. Optimally this value is high.

Next, a statistic is determined. Suppose that RXX is the `r.squared` retrieved from regressing the silhouette widths of object X versus the corresponding cluster membership of object X and RXY the `r.squared` retrieved from regressing the silhouette widths of object X versus the cluster membership determined by object Y and vice versa. The statistic is obtained as:

$$Stat = abs(\sum RXX - \sum RXY)$$

The lower the statistical value, the better the clustering is explained by the sources. Via bootstrapping a p-value is obtained.

Value

A plots are made of the density of the statistic under the null hypotheses. The p-value is also indicated on this plot. Further, a list with two elements is returned:

`Observed Statistic`
: The observed statistical value

`P-Value`
: The P-value of the obtained statistic retrieved after bootstrapping

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)

List=list(fingerprintMat,targetMat)

Comparison=CompareSilCluster(List=List,type="data",
distmeasure=c("tanimoto","tanimoto"),normalize=c(FALSE,FALSE),method=c(NULL,NULL),
nrclusters=7,names=NULL,nboot=100,plottype="new",location=NULL)

Comparison

## End(Not run)
```

`CompareSvsM` *Comparison of clustering results for the single and multiple source clustering.*

Description

A visual comparison of all methods is handy to see which objects will always cluster together independent of the applied methods. The function `CompareSvsM` plots the `ComparePlot` of the single source clustering results on the left and that of the multiple source clustering results on the right such that a visual comparison is possible.

Usage

```
CompareSvsM(ListS, ListM, nrclusters = NULL, cols = NULL,
  fusionsLogS = FALSE, fusionsLogM = FALSE, weightclustS = FALSE,
  weightclustM = FALSE, namesS = NULL, namesM = NULL, margins = c(8.1,
  3.1, 3.1, 4.1), plottype = "new", location = NULL)
```

Arguments

ListS	A list of the outputs from the single source clusterings to be compared. The first element of the list will be used as the reference in ReorderToReference.
ListM	A list of the outputs from the multiple source clusterings to be compared. The first element of the list will be used as the reference.
nrclusters	The number of clusters to cut the dendrogram in. Default is NULL.
cols	A character vector with the names of the colours. Default is NULL.
fusionsLogS	The fusionslog parameter for the elements in ListS. To be handed to ReorderToReference. Default is FALSE.
fusionsLogM	The fusionsLog parameter for the elements in ListM. To be handed to ReorderToReference. Default is FALSE.
weightclustS	The weightclust parameter for the elements in ListS. To be handed to ReorderToReference. Default is FALSE.
weightclustM	The weightclust parameter for the elements in ListM. To be handed to ReorderToReference. Default is FALSE.
namesS	Optional. Names of the single source clusterings to be used as labels for the columns. Default is NULL.
namesM	Optional. Names of the multiple source clusterings to be used as labels for the columns. Default is NULL.
margins	Optional. Margins to be used for the plot. Default is c(8.1,3.1,3.1,4.1).
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Value

The returned value is a plot with on the left the comparison over the objects in ListS and on the right a comparison over the objects in ListM.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
```

```
data(Colors1)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(fingerprintMat,targetMat)

MCF7_W=WeightedClust(List=L,type="data", distmeasure=c("tanimoto","tanimoto"),
normalize=c(FALSE,FALSE),method=c(NULL,NULL),weight=seq(1,0,-0.1),weightclust=0.5
,clust="agnes",linkage="ward",StopRange=FALSE)

ListM=list(MCF7_W)
namesM=seq(1.0,0.0,-0.1)

ListS=list(MCF7_F,MCF7_T)
namesS=c("FP","TP")

CompareSvsM(ListS,ListM,nrclusters=7,cols=Colors1,fusionsLogS=FALSE,
fusionsLogM=FALSE,weightclustS=FALSE,weightclustM=FALSE,namesS,
namesM,plottype="new",location=NULL)

## End(Not run)
```

`ConsensusClustering` *Consensus clustering*

Description

The `ConsensusClustering` includes the ensemble clustering methods IVC, IPVC and IVC which are voting-based consensus methods.

Usage

```
ConsensusClustering(List, type = c("data", "dist", "clust"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), clust = "agnes", linkage = c("flexible",
  "flexible"), alpha = 0.625, nrclusters = c(7, 7), gap = FALSE,
  maxK = 15, votingMethod = c("IVC", "IPVC", "IPC"), optimalk = 7)
```

Arguments

`List`	A list of data matrices. It is assumed the rows are corresponding with the objects.
`type`	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".

distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
nrclusters	The number of clusters to divide each individual dendrogram in. Default is c(7,7) for two data sets.
gap	Logical. Whether the optimal number of clusters should be determined with the gap statistic. Defaults to FALSE.
maxK	The maximal number of clusters to investigate in the gap statistic. Default is 15.
votingMethod	The method to be performed: "IVC", "IPVC","IVC".
optimalk	An estimate of the final optimal number of clusters. Default is 7.

Details

(Nguyen and Caruana 2007) propose three EM-like consensus clustering algorithms: Iterative Voting Consensus (IVC), Iterative Probabilistic Voting Consensus (IPVC) and Iterative Pairwise Consensus (IPC). Given a number of clusters k, the methods iteratively compute the cluster centers and reassign each object to the closest center. IVC and IPVC represent the cluster centers by a vector of the majority votes of the cluster labels of all points belonging to the cluster in each partition. For the reassignment, IVC uses the Hamming distance to compute the distance between the data points and the cluster centers. IPVC is a refinement of IVC as the distance function takes into account the proportion that each feature of a point differs from the points in the cluster. The IPC algorithms is slightly different since the original clusters are built from a similarity matrix which represents the ratio of the number of partitions in which two objects reside in the same cluster. The distance between a data point and a cluster center is the average of the similarity values between the data point and the points residing in the cluster. The iteration ends when the consensus partition does not change.

Value

The returned value is a list of two elements:

DistM	A NULL object
Clust	The resulting clustering

The value has class 'Ensemble'.

References

Anonymous (ed.) (2007). *Consensus clusterings*.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_IVC=ConsensusClustering(List=L,type="data",distmeasure=c("tanimoto", "tanimoto"),
normalize=c(FALSE,FALSE),method=c(NULL,NULL),clust="agnes",linkage = c("flexible",
"flexible"),alpha=0.625,nrclusters=c(7,7),gap = FALSE, maxK = 15,
votingMethod="IVC",optimalk=7)
```

ContFeaturesPlot *Plot of continuous features*

Description

The function ContFeaturesPlot plots the values of continuous features. It is possible to separate between objects of interest and the other objects.

Usage

```
ContFeaturesPlot(leadCpds, data, nrclusters = NULL, orderLab = NULL,
  colorLab = NULL, cols = NULL, ylab = "features", addLegend = TRUE,
  margins = c(5.5, 3.5, 0.5, 8.7), plottype = "new", location = NULL)
```

Arguments

leadCpds	A character vector containing the objects one wants to separate from the others.
data	The data matrix.
nrclusters	Optional. The number of clusters to consider if colorLab is specified. Default is NULL.
orderLab	Optional. If the objects are to set in a specific order of a specific method. Default is NULL.
colorLab	The clustering result that determines the color of the labels of the objects in the plot. If NULL, the labels are black. Default is NULL.
cols	The colors for the labels of the objects. Default is NULL.
ylab	The lable of the y-axis. Default is "features".
addLegend	Logical. Indicates whether a legend should be added to the plot. Default is TRUE.
margins	Optional. Margins to be used for the plot. Default is c(5.5,3.5,0.5,8.7).

plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Value

A plot in which the values of the features of the leadCpds are separeted from the others.

Examples

```
## Not run:
data(Colors1)
Comps=c("Cpd1", "Cpd2", "Cpd3", "Cpd4", "Cpd5")

Data=matrix(sample(15, size = 50*5, replace = TRUE), nrow = 50, ncol = 5)
colnames(Data)=colnames(Data, do.NULL = FALSE, prefix = "col")
rownames(Data)=rownames(Data, do.NULL = FALSE, prefix = "row")
for(i in 1:50){
rownames(Data)[i]=paste("Cpd",i,sep="")
}

ContFeaturesPlot(leadCpds=Comps,orderLab=rownames(Data),colorLab=NULL,data=Data,
nrclusters=7,cols=Colors1,ylab="features",addLegend=TRUE,margins=c(5.5,3.5,0.5,8.7),
plottype="new",location=NULL)

## End(Not run)
```

CVAA *Cumulative voting-based aggregation algorithm*

Description

The CVAA includes the ensemble clustering methods CVAA and W-CVAA which are voting-based consensus methods.

Usage

```
CVAA(Reference = NULL, nrclustersR = 7, List, typeL = c("data", "dist",
  "clust"), distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE,
  FALSE), method = c(NULL, NULL), clust = "agnes", linkage = c("flexible",
  "flexible"), alpha = 0.625, nrclusters = c(7, 7), gap = FALSE,
  maxK = 15, votingMethod = c("CVAA", "W-CVAA"), optimalk = nrclustersR)
```

Arguments

Reference	The reference structure to be updated.
nrclustersR	The number of clusters present in the reference structure. Default is 7.
List	A list of data matrices. It is assumed the rows are corresponding with the objects.
typeL	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
nrclusters	The number of clusters to divide each individual dendrogram in. Default is c(7,7) for two data sets.
gap	Logical. Whether the optimal number of clusters should be determined with the gap statistic. Defaults to FALSE.
maxK	The maximal number of clusters to investigate in the gap statistic. Default is 15.
votingMethod	The method to be performed: "CVAA","W-CVAA".
optimalk	An estimate of the final optimal number of clusters. Default is nrclustersR.

Details

(Saeed, Salim, and Abdo 2012) describe the Cumulative Voting-based Aggregation Algorithm (CVAA) and introduce a variation Weighted Cumulative Voting-based Aggregation Algorithm (W-CVAA, (Saeed, Ahmed, and Shamsir 2014)). In the CVAA algorithm, one data partitioning is chosen as the reference partition. In a first step each partition is relabelled with respect to the reference partition in search of an optimal relabelling. In the next step a consensus partition is obtained. The W-CVAA algorithm is similar but appoints weights to each individual partition. The weights are based on the mutual information of the partition measured by the Shannon entropy.

Value

The returned value is a list of two elements:

DistM	A NULL object
Clust	The resulting clustering

The value has class 'Ensemble'.

References

Saeed F, Salim N and Abdo A (2012). "Voting-based consensus clustering for combining multiple clustering of chemical structures." *Journal of Cheminformatics*, **4**, pp. 37. Saeed F, Ahmed A and Shamsir MS (2014). "Weighted voting-based consensus clustering for chemical structure databases." *Jounral of computer-aided molecular design*, **28**, pp. 675-684.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

MCF7_CVAA=CVAA(Reference=MCF7_T,nrclustersR=7,List=L,typeL="data",
distmeasure=c("tanimoto", "tanimoto"),normalize=c(FALSE,FALSE),method=
c(NULL,NULL),clust="agnes",linkage = c("flexible","flexible"),alpha=0.625,
nrclusters=c(7,7),gap = FALSE, maxK = 15,votingMethod="CVAA",optimalk=7)
```

DetermineWeight_SilClust

Determines an optimal weight for weighted clustering by silhouettes widths.

Description

The function DetermineWeight_SilClust determines an optimal weight for weighted similarity clustering by calculating silhouettes widths. See "Details" for a more elaborate description.

Usage

```
DetermineWeight_SilClust(List, type = c("data", "dist", "clusters"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), weight = seq(0, 1, by = 0.01),
  nrclusters = NULL, names = NULL, nboot = 10, StopRange = FALSE,
  plottype = "new", location = NULL)
```

Arguments

List	A list of matrices of the same type. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").

normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
weight	Optional. A list of different weight combinations for the data sets in List. If NULL, the weights are determined to be equal for each data set. It is further possible to fix weights for some data matrices and to let it vary randomly for the remaining data sets. Defaults to seq(1,0,-0.1). An example is provided in the details.
nrclusters	The number of clusters to cut the dendrogram in. This is necessary for the computation of the Jaccard coefficient. Default is NULL.
names	The labels to give to the elements in List. Default is NULL.
nboot	Number of bootstraps to be run. Default is 10.
StopRange	Logical. Indicates whether the distance matrices with values not between zero and one should be standardized to have so. If FALSE the range normalization is performed. See Normalization. If TRUE, the distance matrices are not changed. This is recommended if different types of data are used such that these are comparable. Default is FALSE.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is FALSE.

Details

For each given weight, a linear combination of the distance matrices of the single data sources is obtained. For these distance matrices, medoid clustering with nrclusters is set up by the pam function of the **cluster** and the silhouette widths are retrieved. These widths indicates how well an object fits in its current cluster. Values around one indicate an appropriate cluster. The silhouette widths are regressed in function of the cluster membership determined by the objects. First, in function of the cluster membership determined by the weighted combination. Then, also in function of the cluster membership determined by the single source clustering. The regression function is fit by the lm function and the r.squared value is retrieved. Ther.squared value indicates how much of the variance of the silhouette widths is explained by the membership. Optimally this value is high.

Next, a statistic is determined. Suppose that RWW is the r.squared retrieved from regressing the weighted silhouette widths versus the weighted cluster membership and RWX the r.squared retrieved from regressing the weighted silhouette widths versus the cluster membership determined by data X. If M is total number of data sources, than statistic is obtained as:

$$Stat = abs(M * RWW - \sum RWX)$$

The lower the statistical value, the better the weighted clustering is explained by the single data sources. The goal is to obtain the weights for which this value is minimized. Via bootstrapping a p-value is obtained for every statistic.

The weight combinations should be provided as elements in a list. For three data matrices an example could be: weights=list(c(0.5,0.2,0.3),c(0.1,0.5,0.4)). To provide a fixed weight for some data sets and let it vary randomly for others, the element "x" indicates a free parameter. An example is weights=list(c(0.7,"x","x")). The weight 0.7 is now fixed for the first data matrix while the remaining 0.3 weight will be divided over the other two data sets. This implies that every combination of the sequence from 0 to 0.3 with steps of 0.1 will be reported and clustering will be performed for each.

Value

Two plots are made: one of the statistical values versus the weights and one of the p-values versus the weights. Further, a list with two elements is returned:

`Result`	A data frame with the statistic for each weight combination
`Weight`	The optimal weight

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)

MC7_Weight=DetermineWeight_SilClust(List=L,type="clusters",distmeasure=
c("tanimoto","tanimoto"),normalize=c(FALSE,FALSE),method=c(NULL,NULL),
weight=seq(0,1,by=0.01),nrclusters=c(7,7),names=c("FP","TP"),nboot=10,
StopRange=FALSE,plottype="new",location=NULL)

## End(Not run)
```

`DetermineWeight_SimClust` *Determines an optimal weight for weighted clustering by similarity weighted clustering.*

Description

The function DetermineWeight_SimClust determines an optimal weight for performing weighted similarity clustering on by applying similarity clustering. For each given weight, is each separate clustering compared to the clustering on a weighted dissimilarity matrix and a Jaccard coefficient is calculated. The ratio of the Jaccard coefficients closets to one indicates an optimal weight.

Usage

```
DetermineWeight_SimClust(List, type = c("data", "dist", "clusters"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), weight = seq(0, 1, by = 0.01),
  nrclusters = NULL, clust = "agnes", linkage = c("flexible", "flexible"),
  linkageF = "ward", alpha = 0.625, gap = FALSE, maxK = 15,
  names = NULL, StopRange = FALSE, plottype = "new", location = NULL)
```

Arguments

List	A list of matrices of the same type. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
weight	Optional. A list of different weight combinations for the data sets in List. If NULL, the weights are determined to be equal for each data set. It is further possible to fix weights for some data matrices and to let it vary randomly for the remaining data sets. Defaults to seq(1,0,-0.1). An example is provided in the details.
nrclusters	The number of clusters to cut the dendrogram in. This is necessary for the computation of the Jaccard coefficient. Default is NULL.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for the individual clusterings. Defaults to c("flexible","flexible").
linkageF	Choice of inter group dissimilarity (character) for the final clustering. Defaults to "ward".
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible".

gap	Logical. Whether or not to calculate the gap statistic in the clustering on each data matrix separately. Only if type="data". Default is FALSE.
maxK	The maximal number of clusters to consider in calculating the gap statistic. Only if type="data". Default is 15.
names	The labels to give to the elements in List. Default is NULL.
StopRange	Logical. Indicates whether the distance matrices with values not between zero and one should be standardized to have so. If FALSE the range normalization is performed. See Normalization. If TRUE, the distance matrices are not changed. This is recommended if different types of data are used such that these are comparable. Default is FALSE.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is FALSE.

Details

If the type of List is data, an hierarchical clustering is performed on each data matrix separately. After obtaining clustering results for the two data matrices, the distance matrices are extracted. If these are not calculated with the same distance measure, they are normalized to be in the same range. For each weight, a weighted linear combination of the distance matrices is taken and hierarchical clustering is performed once again. The resulting clustering is compared to each of the separate clustering results and a Jaccard coefficient is computed. The ratio of the Jaccard coefficients closets to one, indicates an optimal weight. A plot of all the ratios is produced with an extra indication for the optimal weight.

The weight combinations should be provided as elements in a list. For three data matrices an example could be: weights=list(c(0.5,0.2,0.3),c(0.1,0.5,0.4)). To provide a fixed weight for some data sets and let it vary randomly for others, the element "x" indicates a free parameter. An example is weights=list(c(0.7,"x","x")). The weight 0.7 is now fixed for the first data matrix while the remaining 0.3 weight will be divided over the other two data sets. This implies that every combination of the sequence from 0 to 0.3 with steps of 0.1 will be reported and clustering will be performed for each.

Value

The returned value is a list with three elements:

ClustSep	The result of Cluster for each single element of List
Result	A data frame with the Jaccard coefficients and their ratios for each weight
Weight	The optimal weight

References

Perualila-Tan N, Shkedy Z, Talloen W, Goehlmann HWH, Consortium Q, Van Moerbeke M and Kasim A (2016). "Weighted-Similarity Based Clustering of Chemical Structure and Bioactivity Data in Early Drug Discovery." *Journal of Bioinformatics and Computational Biology*, **14**(4), pp. 1650018.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",alpha=0.625,gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",alpha=0.625,gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)

MCF7_Weight=DetermineWeight_SimClust(List=L,type="clusters",weight=seq(0,1,by=0.01),
nrclusters=c(7,7),distmeasure=c("tanimoto","tanimoto"),normalize=c(FALSE,FALSE),
method=c(NULL,NULL),clust="agnes",linkage=c("flexible","flexible"),linkageF="ward",
alpha=0.625,gap=FALSE,maxK=50,names=c("FP","TP"),StopRange=FALSE,plottype="new",location=NULL)

## End(Not run)
```

DiffGenes *Differential gene expressions for multiple results*

Description

The function DiffGenes will, given the output of a certain method, look for genes that are differentially expressed for each cluster by applying the limma function to that cluster and compare it to all other clusters simultaneously. If a list of outputs of several methods is provided, DiffGenes will perform the limma function for each method.

Usage

```
DiffGenes(List, Selection = NULL, geneExpr = NULL, nrclusters = NULL,
  method = "limma", sign = 0.05, topG = NULL, fusionsLog = TRUE,
  weightclust = TRUE, names = NULL)
```

Arguments

List	A list of the clustering outputs to be compared. The first element of the list will be used as the reference in ReorderToReference.

Selection	If differential gene expression should be investigated for a specific selection of objects, this selection can be provided here. Selection can be of the type "character" (names of the objects) or "numeric" (the number of specific cluster). Default is NULL.
geneExpr	The gene expression matrix or ExpressionSet of the objects. The rows should correspond with the genes.
nrclusters	Optional. The number of clusters to cut the dendrogram in. The number of clusters should not be specified if the interest lies only in a specific selection of objects which is known by name. Otherwise, it is required. Default is NULL.
method	The method to applied to look for DE genes. For now, only the limma method is available. Default is "limma".
sign	The significance level to be handled. Default is 0.05.
topG	Overrules sign. The number of top genes to be shown. Default is NULL.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE
weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
names	Optional. Names of the methods. Default is NULL.

Details

The function rearranges the clusters of the methods to a reference method such that a comparison is made easier. Given a list of methods, it calls upon ReorderToReference to rearrange the number of clusters according to the first element of the list which will be used as the reference.

Value

The returned value is a list with an element per method. Each element contains a list per cluster with the following elements:

objects	A list with the elements LeadCpds (the objects of interest) and OrderedCpds (all objects in the order of the clustering result)
Genes	A list with the elements TopDE (a table with information on the top genes) and AllDE (a table with information on all genes)

References

SMYTH, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology. 3(1).

Examples

```
data(fingerprintMat)
data(targetMat)
```

```
data(geneMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_T ,MCF7_F)

MCF7_FT_DE = DiffGenes(List=L,geneExpr=geneMat,nrclusters=7,method="limma",
sign=0.05,topG=10,fusionsLog=TRUE,weightclust=TRUE)
```

DiffGenesSelection *Differential expression for a selection of objects*

Description

Internal function of DiffGenes.

Usage

```
DiffGenesSelection(List, Selection, geneExpr = NULL, nrclusters = NULL,
  method = "limma", sign = 0.05, topG = NULL, fusionsLog = TRUE,
  weightclust = TRUE, names = NULL)
```

Arguments

List	A list of the clustering outputs to be compared. The first element of the list will be used as the reference in ReorderToReference.
Selection	If differential gene expression should be investigated for a specific selection of objects, this selection can be provided here. Selection can be of the type "character" (names of the objects) or "numeric" (the number of specific cluster). Default is NULL.
geneExpr	The gene expression matrix or ExpressionSet of the objects. The rows should correspond with the genes.
nrclusters	Optional. The number of clusters to cut the dendrogram in. The number of clusters should not be specified if the interest lies only in a specific selection of objects which is known by name. Otherwise, it is required. Default is NULL.
method	The method to applied to look for DE genes. For now, only the limma method is available. Default is "limma".
sign	The significance level to be handled. Default is 0.05.
topG	Overrules sign. The number of top genes to be shown. Default is NULL.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE

weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
names	Optional. Names of the methods. Default is NULL.

Distance *Distance calculation*

Description

The Distance function calculates the distances between the data objects. The included distance measures are euclidean for continuous data and the tanimoto coefficient or jaccard index for binary data.

Usage

```
Distance(Data, distmeasure = c("tanimoto", "jaccard", "euclidean", "hamming",
  "cont tanimoto", "MCA_coord", "gower", "chi.squared", "cosine"),
  normalize = FALSE, method = NULL)
```

Arguments

Data	A data matrix. It is assumed the rows are corresponding with the objects.
distmeasure	Choice of metric for the dissimilarity matrix (character). Should be one of "tanimoto", "euclidean", "jaccard","hamming","cont tanimoto","MCA_coord","gower","chi.squared" or "cosine"
normalize	Logical. Indicates whether to normalize the distance matrices or not, default is FALSE. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is NULL.

Details

The euclidean distance distance is included for continuous matrices while for binary matrices, one has the choice of either the jaccard index, the tanimoto coeffcient or the hamming distance. The hamming distance is obtained by applying the hamming.distance function of the **e1071** package. It will compute the hamming distance between the rows of the data matrix. The hamming distance counts the number of times where two rows differ in their zero and one values. The Jaccard index is calcaluted as determined by the formula of the dist.binary function in the **a4** package and the tanimoto coefficient as described by *Li2011*. For both, first the similarity is calculated as

$$s = fracn11n11 + n10 + n01$$

with n11 the number of features the 2 objects have in common, n10 the number of features of the first compound and n01 the number of features of the second compound. These similarities are converted to distances by:

$$J = \sqrt{1 - s}$$

for the jaccard index and by:

$$T = 1 - s$$

for the tanimoto coefficient. The lower the similarity values s are, the more features are shared between the two objects and the more alike they are. Since clustering is based on dissimilarity, the conversion to distances is performed. If normalize=TRUE and the distance meausure is euclidean, the data matrix is normalized beforehand. Further, a version of the tanimoto coefficient is also available for continuous data.

Value

The returned value is a distance matrix.

Examples

```
data(fingerprintMat)
Dist_F=Distance(fingerprintMat,distmeasure="tanimoto",normalize=FALSE,method=NULL)
```

---

distanceheatmaps *Determine the distance in a heatmap*

Description

Internal function of HeatmapPlot

Usage

```
distanceheatmaps(Data1, Data2, names = NULL, nrclusters = 7)
```

Arguments

Data1	The resulting clustering of method 1.
Data2	The resulting clustering of method 2.
names	The names of the objects in the data sets. Default is NULL.
nrclusters	The number of clusters to cut the dendrogram in. Default is NULL.

EHC *Ensemble for hierarchical clustering*

Description

The Ensemble for Hierarchical Clustering (EHC, (Hossain, Bridges, Wang, and Hodges 2012)) defines the strength of association between a pair of objects as a measure of how closely these are associated taking into account the levels of the dendrogram. Therefore, the sum of the normalized depths of the clusters in which both objects reside is taken as a measure of association. The depths are weighted by the intra-cluster proximity values. The resulting similarity matrix is seen as an adjacency matrix of a graph and the METIS algorithm is performed to cut the graph in k clusters.

Usage

```
EHC(List, type = c("data", "dist", "clust"), distmeasure = c("tanimoto",
  "tanimoto"), normalize = c(FALSE, FALSE), method = c(NULL, NULL),
  clust = "agnes", linkage = c("flexible", "flexible"), alpha = 0.625,
  gap = FALSE, maxK = 15, graphPartitioning = c("METIS", "MST"),
  optimalk = 7, waitingtime = 300, file_number = 0, executable = FALSE)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or"clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, default is FALSE. This is recommended if different distance types are used. More details on normalization in Normalization
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
gap	Logical. Whether the optimal number of clusters should be determined with the gap statistic. Defaults to FALSE.
maxK	The maximal number of clusters to investigate in the gap statistic. Default is 15.

`graphPartitioning`
: The graph-partitioning method to be performed: "METIS" (implemented in MATLAB), "MST".

`optimalk`
: An estimate of the final optimal number of clusters. Default is 7.

`waitingtime`
: The time in seconds to wait until the MATLAB results are generated. Defaults to 300.

`file_number`
: The specific file number to be placed as a tag in the file generated by MATLAB. Defaults to 00.

`executable`
: Logical. Whether the METIS MATLAB function is performed via an executable on the command line (TRUE, only possible for Linux systems) or by calling on MATLAB directly (FALSE). Defaults to FALSE.

Value

The returned value is a list of two elements:

`DistM`
: The resulting distance matrix

`Clust`
: The resulting clusters

The value has class 'Ensemble'.

References

Hossain M, Bridges SM, Wang Y and Hodges JE (2012). "An effective ensemble method for hierarchical clustering." In *Proceedings of the Fifth International C* Conference on Computer Science and Software Engineering*, pp. 18-26.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_EHC=EHC(List=L,type="data",distmeasure=c("tanimoto", "tanimoto"),normalize=
c(FALSE,FALSE),method=c(NULL,NULL),clust="agnes",linkage = c("flexible","flexible"),
alpha=0.625,gap=FALSE,maxK=15,graphPartitioning="METIS",optimalk=7,
waitingtime=300,file_number=00,executable=FALSE)


## End(Not run)
```

EnsembleClustering *Ensemble clustering*

Description

The EnsembleClustering includes the ensemble clustering methods CSPA, HGPA and MCLA which are graph-based consensus methods.

Usage

```
EnsembleClustering(List, type = c("data", "dist", "clust"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), clust = "agnes", linkage = c("flexible",
  "flexible"), alpha = 0.625, nrclusters = 7, gap = FALSE, maxK = 15,
  ensembleMethod = c("CSPA", "HGPA", "MCLA", "Best"), waitingtime = 300,
  file_number = 0, executable = FALSE)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible,", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
nrclusters	The number of clusters to divide each individual dendrogram in. Default is c(7,7) for two data sets.
gap	Logical. Whether the optimal number of clusters should be determined with the gap statistic. Default is FALSE.
maxK	The maximal number of clusters to investigate in the gap statistic. Default is 15.
ensembleMethod	The method to be performed: "CSPA", "HGPA", "MCLA" or "Best".

`waitingtime` The time in seconds to wait until the MATLAB results are generated. Defaults to 300.

`file_number` The specific file number to be placed as a tag in the file generated by MATLAB. Defaults to 00.

`executable` Logical. Whether the MATLAB functions are performed via an executable on the command line (TRUE, only possible for Linux systems) or by calling on MATLAB directly (FALSE). Defaults to FALSE. The files EnsembleClusteringC.m (CSPA), EnsembleClusteringH.m (HGPA), EnsembleClusteringM.m (MCLA) and MetisAlgorithm.m are present in the inst folder to be transformed in executables.

Details

(Strehl and Gosh 2002) introduce three heuristic algorithms to solve the cluster ensemble problem. Each method starts by transforming the clustering solutions into a single hypergraph in which a hyperedge represents a single cluster. The Cluster-based Similarity Partitioning Algorithm (CSPA) transforms the hypergraph into an overall similarity matrix which entries represent the fraction of clusterings in which two objects are in the same cluster. The similarity matrix is considered as a graph and the objects are reclustered with the graph partitioning algorithm METIS (Karypis and Kumar 1998). Hyper-Graph Partitioning Algorithm (HGPA) partitions the hypergraph directly by cutting a minimal number of hyperedges. It aims to obtain connected components of approximately the same dimension. The partitioning algorithm is HMetis (Karypis, Aggarwal, Kumar, and Shekhar 1997). The Meta-CLustering Algorithm (MCLA) computes a similarity between the hyperedges (clusters) based on the Jaccard index. The resulting similarity matrix is used to build a meta-graph which is partitioned by the METIS algorithm Karypis G and Kumar V (1998). "A fast and high quality multilevel scheme for partitioning irregular graphs." *SIAM*, **20**, pp. 359-392. into resulting meta-clusters. The final partition of the objects is obtaining by appointing each object to the meta-cluster to which it is assigned the most. The `R` code calls on the MATLAB code provided by (Strehl and Gosh 2002). The MATLAB functions are included in the inst folder and should be located in the working directory. Shell script for the executable can be found in the inst folder as well.

Value

The returned value is a list of two elements:

`DistM` A list with the distance matrix for each data structure

`Clust` The resulting clustering

The value has class 'Ensemble'.

References

Strehl A and Gosh J (2002). "Cluster ensembles - A knowledge reuse framework for combining multiple partitions." *Journal of Machine Learning Research*, **3**, pp. 583-617. Karypis G, Aggarwal R, Kumar V and Shekhar S (1997). "Multilevel hypergraph partitioning: Application in VLSI domain." In *In Proceedings of the 34th annual Design Automation Conference*, series ACM, pp. 526-529. Karypis G and Kumar V (1998). "A fast and high quality multilevel scheme for partitioning irregular graphs." *SIAM*, **20**, pp. 359-392.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_CSPA=EnsembleClustering(List=L,type="data",distmeasure=c("tanimoto",
"tanimoto"),normalize=c(FALSE,FALSE),method=c(NULL,NULL),StopRange=FALSE,
clust="agnes",linkage=c("flexible","flexible"),nrclusters=c(7,7),gap=FALSE,
maxK=15,ensembleMethod="CSPA",executable=FALSE)


## End(Not run)
```

`EvidenceAccumulation` *Evidence accumulation*

Description

The Evidence Accumulation (EA, (Fred and Jain 2002)) sets up the same similarity matrix and applies a minimum spanning tree (MST,(Prim 1957)) algorithm to the corresponding graph. The algorithm will connect all objects with the shortest past. In order to recover the clusters, weak edges are cut at a threshold value t.

Usage

```
EvidenceAccumulation(List, type = c("data", "dist", "clust"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), clust = "agnes", linkage = c("flexible",
  "flexible"), alpha = 0.625, nrclusters = c(7, 7), gap = FALSE,
  maxK = 15, graphPartitioning = c("MTS", "SL", "SL_agnes"), t = NULL)
```

Arguments

`List`	A list of data matrices. It is assumed the rows are corresponding with the objects.
`type`	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or"clusters".
`distmeasure`	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
`normalize`	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in `Normalization`.
`method`	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.

clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
nrclusters	The number of clusters to divide each individual dendrogram in. Default is c(7,7) for two data sets.
gap	Logical. Whether the optimal number of clusters should be determined with the gap statistic. Defaults to FALSE.
maxK	The maximal number of clusters to investigate in the gap statistic. Default is 15.
graphPartitioning	The method to be performed: "MTS", "SL", "SL_agnes".
t	A threshold to cut weak edges. Default is NULL.

Value

The returned value is a list of two elements:

DistM	A NULL object
Clust	The resulting clustering

The value has class 'Ensemble'.

References

Fred ALN and Jain AK (2002). "Data clustering using evidence accumulation." *International Conference on Pattern Recognition.*, **16**, pp. 276- 280 vol.4. Prim R (1957). "Shortest connection networks and some generalizations." *Bell System Technical Journal*, **37**, pp. 1389-1401.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_EA=EvidenceAccumulation(List=L,type="data",distmeasure=c("tanimoto", "tanimoto"),
normalize=c(FALSE,FALSE),method=c(NULL,NULL),clust="agnes",linkage = c("flexible",
"flexible"),alpha=0.625,nrclusters=c(7,7),gap = FALSE, maxK = 15,graphPartitioning="MTS")
```

f.clustABC.MultiSource *f.clustABC.MultiSoucre*

Description

Internal function of M_ABC: performs the final clustering.

Usage

```
f.clustABC.MultiSource(res, numclust = NULL, distmeth = 1,
  linkage = "ward", alpha = 0.625, mds = FALSE)
```

Arguments

res	Matrix object whose rows are the base clusters determined in each iteration of the ABC algorithm.
numclust	The number of clusters. Default is NULL.
distmeth	Binary, indicating whether the ABC dissimilarities should be scaled by the total number of simulations. By default distmeth=1 indicating the dissimilarities should be left as is.
linkage	Choice of inter group dissimilarity (character) for the final clustering. Defaults to "ward.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
mds	Logical, indicating whether and MDS plot of the dissimilarities should be drawn. Default is FALSE

f.gsample *f.gsample*

Description

internal function of M_ABC: samples a predetermined number of rows according to specified weights.

Usage

```
f.gsample(zf, ng = 100)
```

Arguments

zf	Statistics to be used in the weighting of the rows.
ng	The number to resample. Default is 100.

f.rmv *f.rmv*

Description

internal function of M_ABC: computes means and variances.

Usage

```
f.rmv(x, varonly = FALSE)
```

Arguments

x	data variable
varonly	Logical. Whether only the variance should be returned (TRUE) or the mean as well (FALSE). Default is FALSE.

f.t *ff*

Description

Internal function of M_ABC: determine weights with two sample t-test.

Usage

```
f.t(x, his = FALSE)
```

Arguments

x	data variable
his	Logical. Whether a histogram should be plotted. Default is FALSE.

FeatSelection *feature selection for a selection of objects*

Description

Internal function of CharacteristicFeatures.

Usage

```
FeatSelection(List, Selection = NULL, binData = NULL, contData = NULL,
  datanames = NULL, nrclusters = NULL, topChar = NULL, sign = 0.05,
  fusionsLog = TRUE, weightclust = TRUE)
```

Arguments

List	A list of the clustering outputs to be compared. The first element of the list will be used as the reference in ReorderToReference.
Selection	If differential gene expression should be investigated for a specific selection of objects, this selection can be provided here. Selection can be of the type "character" (names of the objects) or "numeric" (the number of specific cluster). Default is NULL.
binData	A list of the binary feature data matrices. These will be evaluated with the fisher's extact test. Default is NULL.
contData	A list of continuous data sets of the objects. These will be evaluated with the t-test. Default is NULL.
datanames	A vector with the names of the data matrices. Default is NULL.
nrclusters	Optional. The number of clusters to cut the dendrogram in. The number of clusters should not be specified if the interest lies only in a specific selection of objects which is known by name. Otherwise, it is required. Default is NULL.
topChar	Overrules sign. The number of features to display for each cluster. If not specified, only the significant genes are shown. Default is NULL.
sign	The significance level to be handled. Default is 0.05.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE
weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.

`FeaturesOfCluster` *List all features present in a selected cluster of objects*

Description

The function `FeaturesOfCluster` lists the number of features objects of the cluster have in common. A threshold can be set selecting among how many objects of the cluster the features should be shared. An optional plot of the features is available.

Usage

```
FeaturesOfCluster(leadCpds, data, threshold = 1, plot = TRUE,
  plottype = "new", location = NULL)
```

Arguments

`leadCpds`	A character vector containing the objects one wants to investigate in terms of features.
`data`	The data matrix.
`threshold`	The number of objects the features at least should be shared amongst. Default is set to 1 implying that the features should be present in at least one of the objects specified in leadCpds.
`plot`	Logical. Indicates whether or not a BinFeaturesPlot should be set up for the selectcion of objects and discovered features. Default is FALSE.
`plottype`	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
`location`	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Value

A plot indicating the values of the features of the LeadCpds in green and those of the others in blue. It list all features which are present in at least the threshold number of objects. By including all other objects as well, one can see whether features are common in the objects or rather specific for the cluster.

Further, it returns a list with 2 items. The first indicates the number of shared features among the objects. This provides an overview of which objects are more similar than others. The second item is a character vector of the plotted features such that these can be retrieved for further investigation.

Examples

```
## Not run:
data(fingerprintMat)

Lead=rownames(fingerprintMat)[1:5]

FeaturesOfCluster(leadCpds=Lead,data=fingerprintMat,
threshold=1,plot=TRUE,plottype="new",location=NULL)

## End(Not run)
```

FindCluster *Find a selection of objects in the output of* ReorderToReference

---

Description

FindCluster selects the objects belonging to a cluster after the results of the methods have been rearranged by the ReorderToReference.

Usage

```
FindCluster(List, nrclusters = NULL, select = c(1, 1), fusionsLog = TRUE,
  weightclust = TRUE, names = NULL)
```

Arguments

List	A list of the clustering outputs to be compared. The first element of the list will be used as the reference in ReorderToReference.
nrclusters	The number of clusters to cut the dendrogram in. Default is NULL.
select	The row (the method) and the number of the cluster to select. Default is c(1,1).
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE
weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
names	Optional. Names of the methods. Default is NULL.

Value

A character vector containing the names of the objects in the selected cluster.

Examples

```
data(fingerprintMat)
data(targetMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
names=c("FP","TP")

Comps=FindCluster(List=L,nrclusters=7,select=c(1,4))
Comps
```

FindElement *Find an element in a data structure*

Description

The function FindElement is used internally in the PreparePathway function but might come in handy for other uses as well. Given the name of an object, the function searches for that object in the data structure and extracts it. When multiple objects have the same name, all are extracted.

Usage

```
FindElement(what = NULL, object = NULL, element = list())
```

Arguments

what	A character string indicating which object to look for. Default is NULL.
object	The data structure to look into. Only the classes data frame and list are supported. Default is NULL.
element	Not to be specified by the user.

Value

The returned value is a list with an element for each object found. The element contains everything the object contained in the original data structure.

Examples

```
data(fingerprintMat)
data(targetMat)
data(geneMat)
```

```
MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

MCF7_DiffGenes_FandT10=DiffGenes(list(MCF7_F,MCF7_T),Selection=NULL,geneExpr=geneMat,
nrclusters=7,method="limma",sign=0.05,top=10,fusionsLog = TRUE, weightclust = TRUE,
names = NULL)

Find=FindElement(what='TopDE',object=MCF7_DiffGenes_FandT10)
```

`FindGenes` *Investigates whether genes are differential expressed in multiple clusters*

Description

Due to the shifting of objects over the clusters for the different methods, it is possible that the same gene is found significant for a different cluster in another method. These can be tracked with the `FindGenes` function. Per method and per cluster, it will take note of the genes found significant and investigate if these were also find for another cluster in another method.

Usage

```
FindGenes(dataLimma, names = NULL)
```

Arguments

`dataLimma`	Preferably an output of the `DiffGenes` function. If not, an ID element of the top genes must be present for each cluster of each method specified in the data structure.
`names`	Optional. Names of the methods. Default is NULL.

Value

The returned value is a list with an element per cluster and per cluster one for every gene. Per gene, a vector is given which contain the methods for which the gene was found. If the cluster is changed compared to the reference method of DataLimma, this is indicated with an underscore.

Author(s)

Marijke Van Moerbeke

Examples

```
data(fingerprintMat)
data(targetMat)
data(geneMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

MCF7_DiffGenes_FandT10=DiffGenes(list(MCF7_F,MCF7_T),Selection=NULL,geneExpr=geneMat,
nrclusters=7,method="limma",sign=0.05,top=10,fusionsLog = TRUE, weightclust = TRUE,
names = NULL)

MCF7_SharedGenes=FindGenes(dataLimma=MCF7_DiffGenes_FandT10,names=c("FP","TP"))
```

---

fingerprintMat *Fingerprint data*

Description

A binary data matrix that contains 250 fingerprints for a set of 56 compounds.

Usage

```
data(fingerprintMat)
```

Format

An object of class "matrix".

Examples

```
data(fingerprintMat)
```

GeneInfo *Information of the genes Gene info in a data frame*

Description

Information of the genes

Gene info in a data frame

Format

A data frame with 3 variables: ENTREZID, SYMBOL and GENENAME

Examples

```
data(GeneInfo)
```

---

geneMat *Gene expression data*

Description

Gene expression data for 2434 genes for a set of 56 compounds.

Usage

```
data(geneMat)
```

Format

An object of class "matrix".

Examples

```
data(geneMat)
```

Geneset.intersect *Intersection over resulting gene sets of* PathwaysIter *function*

Description

The function Geneset.intersect collects the results of the PathwaysIter function per method for each cluster and takes the intersection over the iterations per cluster per method. This is to see if over the different resamplings of the data, similar pathways were discovered.

Usage

```
Geneset.intersect(PathwaysOutput, Selection = FALSE, sign = 0.05,
  names = NULL, seperatetables = FALSE, separatepvals = FALSE)
```

Arguments

PathwaysOutput	The output of the PathwaysIter function.
Selection	Logical. Indicates whether or not the output of the pathways function were concentrated on a specific selection of objects. If this was the case, Selection should be put to TRUE. Otherwise, it should be put to FALSE. Default is TRUE.
sign	The significance level to be handled for cutting of the pathways. Default is 0.05.
names	Optional. Names of the methods. Default is NULL.

`seperatetables` Logical. If TRUE, a separate element is created per cluster containing the pathways for each iteration. Default is FALSE.

`separatepvals` Logical. If TRUE, the p-values of the each iteration of each pathway in the intersection is given. If FALSE, only the mean p-value is provided. Default is FALSE.

Value

The output is a list with an element per method. For each method, it is portrayed per cluster which pathways belong to the intersection over all iterations and their corresponding mean p-values.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(geneMat)
data(GeneInfo)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)

MCF7_Paths_FandT=PathwaysIter(List=L, geneExpr = geneMat, nrclusters = 7, method =
c("limma", "MLP"), geneInfo = GeneInfo, geneSetSource = "GOBP", topP = NULL,
topG = NULL, GENESET = NULL, sign = 0.05,niter=2,fusionsLog = TRUE,
weightclust = TRUE, names =names)

MCF7_Paths_intersection=Geneset.intersect(PathwaysOutput=MCF7_Paths_FandT,
sign=0.05,names=c("FP","TP"),seperatetables=FALSE,separatepvals=FALSE)

str(MCF7_Paths_intersection)

## End(Not run)
```

Geneset.intersectSelection

Intersection over resulting gene sets of `PathwaysIter` *function for a selection of objects*

Description

Internal function of `Geneset.intersect`.

Usage

```
Geneset.intersectSelection(list.output, sign = 0.05, names = NULL,
  seperatetables = FALSE, separatepvals = FALSE)
```

Arguments

list.output	The output of the PathwaysIter function.
sign	The significance level to be handled for cutting of the pathways. Default is 0.05.
names	Optional. Names of the methods. Default is NULL.
seperatetables	Logical. If TRUE, a separate element is created per cluster containing the pathways for each iteration. Default is FALSE.
separatepvals	Logical. If TRUE, the p-values of the each iteration of each pathway in the intersection is given. If FALSE, only the mean p-value is provided. Default is FALSE.

GS *List of GO Annotations*

Description

A list that contains the GO annotations produced by getGeneSets of the MLP package for the genes in the geneMat data.

Format

A data frame with 3 variables: ENTREZID, SYMBOL and GENENAME

Examples

```
data(GS)
```

HBGF *Hybrid bipartite graph formulation*

Description

Hybrid Bipartite Graph Formulation (HBGF) is a graph-based consensus multi-source clustering technique. The method builds a bipartite graph in which the two types of vertices are represented by the objects on one hand and the clusters of the partitions on the other hand. An edge is only present between an object vertex and a cluster vertex indicating that the object belongs to that cluster. The graph can be partitioned with the Spectral clustering (Ng, Jordan, and Weiss 2000).

Usage

```
HBGF(List, type = c("data", "dist", "clust"), distmeasure = c("tanimoto",
  "tanimoto"), normalize = c(FALSE, FALSE), method = c(NULL, NULL),
  clust = "agnes", linkage = c("flexible", "flexible"), alpha = 0.625,
  nrclusters = c(7, 7), gap = FALSE, maxK = 15,
  graphPartitioning = "Spec", optimalk = 7)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
nrclusters	The number of clusters to divide each individual dendrogram in. Default is c(7,7) for two data sets.
gap	Logical. Whether the optimal number of clusters should be determined with the gap statistic. Default is FALSE.
maxK	The maximal number of clusters to investigate in the gap statistic. Default is 15.
graphPartitioning	A character string indicating the preferred graph partitioning algorithm. For now only spectral clustering ("Spec") is implemented. Defaults to "Spec".
optimalk	An estimate of the final optimal number of clusters. Default is 7.

Value

The returned value is a list of two elements:

DistM	A NULL object
Clust	The resulting clustering

The value has class 'Ensemble'.

References

Fern XZ and Brodley CE (2004). "Solving cluster ensemble problems by bipartite graph partitioning." In *Proceedings of the 21th International Conference on Machine Learning*.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_HBGF=HBGF(List=L,type="data",distmeasure=c("tanimoto","tanimoto"),normalize=
c(FALSE,FALSE),method=c(NULL,NULL),clust="agnes",linkage = c("flexible",
"flexible"),nrclusters=c(7,7),gap = FALSE, maxK = 15,graphPartitioning="Spec",
optimalk=7)
```

`HeatmapPlot` *Comparing two clustering results with a heatmap*

Description

The `HeatmapCols` function calculates the distance between two outputs of clustering methods and plots the resulting heatmap. The function heatmap.2 is called upon to make the actual plot of the heatmap. It is noted that for thi s function the number of colors should be one more than the number of clusters to color the so called zero cells in the distance matrix.

Another way to compare to methods is via an adaptation of heatmaps. The input of this function is the resulting clustering (the Clust element of the list) of two methods and can be seen as: method 1 versus method 2. The dendrograms are cut into a specific number of clusters. Each cluster of method 2 and its members are given a distinct color represented by a number. These are the clusters to which a comparison is made. A matrix is set up of which the columns are determined by the ordering of clustering of method 2 and the rows by the ordering of method 1. Every column represent one object just as every row and every column represent the color of its cluster. A function visits every cell of the matrix. If the objects represented by the cell are still together in a cluster, the color of the column is passed to the cell. This creates the distance matrix which can be given to the HeatmapCols function to create the heatmap.

Usage

```
HeatmapPlot(Data1, Data2, names = NULL, nrclusters = NULL, cols = NULL,
  plottype = "new", location = NULL)
```

Arguments

`Data1`	The resulting clustering of method 1.
`Data2`	The resulting clustering of method 2.
`names`	The names of the objects in the data sets. Default is NULL.
`nrclusters`	The number of clusters to cut the dendrogram in. Default is NULL.

cols	A character vector with the colours for the clusters. Default is NULL.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Value

A heatmap based on the distance matrix created by the function with the dendrogram of method 2 on top of the plot and the one from method 1 on the left. The names of the objects are depicted on the bottom in the order of clustering of method 2 and on the right by the ordering of method 1. Vertically the cluster of method 2 can be seen while horizontally those of method 1 are portrayed.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(Colors1)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
clust="agnes",linkage="flexible",gap=FALSE,maxK=15)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
clust="agnes",linkage="flexible",gap=FALSE,maxK=15)


L=list(MCF7_F,MCF7_T)
names=c("FP","TP")

HeatmapPlot(Data1=MCF7_T,Data2=MCF7_F,names=rownames(fingerprintMat)
,nrclusters=7,cols=Colors1,plottype="new", location=NULL)

## End(Not run)
```

HeatmapSelection *A function to select a group of objects via the similarity heatmap.*

Description

The function HeatmapSelection plots the similarity values between objects. The plot is similar to the one produced by SimilarityHeatmap but without the dendrograms on the sides. The function is rather explorative and experimental and is to be used with some caution. By clicking in the plot, the user can select a group of objects of interest. See more in Details.

A similarity heatmap is created in the same way as in SimilarityHeatmap. The user is now free to select two points on the heatmap. It is advised that these two points are in opposite corners of a square that indicates a high similarity among the objects. The points do not have to be the exact corners of the group of interest, a little deviation is allowed as rows and columns of the selected subset of the matrix with sum equal to 1 are filtered out. A sum equal to one, implies that the compound is only similar to itself.

The function is meant to be explorative but is experimental. The goal was to make the selection of interesting objects easier as sometimes the labels of the dendrograms are too distorted to be read. If the figure is exported to a pdf file with an appropriate width and height, the labels can be become readable again.

Usage

```
HeatmapSelection(Data, type = c("data", "dist", "clust", "sim"),
  distmeasure = "tanimoto", normalize = FALSE, method = NULL,
  linkage = "flexible", cutoff = NULL, percentile = FALSE,
  dendrogram = NULL, width = 7, height = 7)
```

Arguments

Data	The data of which a heatmap should be drawn.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	The distance measure. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to "tanimoto".
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is NULL.
linkage	Choice of inter group dissimilarity (character). Defaults to "flexible".
cutoff	Optional. If a cutoff value is specified, all values lower are put to zero while all other values are kept. This helps to highlight the most similar objects. Default is NULL.
percentile	Logical. The cutoff value can be a percentile. If one want the cutoff value to be the 90th percentile of the data, one should specify cutoff = 0.90 and percentile = TRUE. Default is FALSE.
dendrogram	Optional. If the clustering results of the data is already available and should not be recalculated, this results can be provided here. Otherwise, it will be calculated given the data. This is necessary to have the objects in their order of clustering on the plot. Default is NULL.
width	The width of the plot to be made. This can be adjusted since the default size might not show a clear picture. Default is 7.
height	The height of the plot to be made. This can be adjusted since the default size might not show a clear picture. Default is 7.

Value

A heatmap with the names of the objects on the right and bottom. Once points are selected, it will return the names of the objects that are in the selected square provided that these show similarity among each other.

Examples

```
## Not run:
data(fingerprintMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55)

HeatmapSelection(Data=MCF7_F$DistM,type="dist",cutoff=0.90,percentile=TRUE,
dendrogram=MCF7_F,width=7,height=7)

## End(Not run)
```

HierarchicalEnsembleClustering

Hierarchical ensemble clustering

Description

(Zheng, Li, and Ding 2014) proposed the Hierarchical Ensemble Clustering (HEC) algorithm. For each dendrogram, the cophenetic distances between the object are calculated. The distances are aggregated across the data sets and an ultra-metric which is the closest to the distance matrix is determined. A final hierarchical clustering is based on the ultra-metric values.

Usage

```
HierarchicalEnsembleClustering(List, type = c("data", "dist", "clust"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), clust = "agnes", linkage = c("flexible",
  "flexible"), alpha = 0.625)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or"clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").

normalize	Logical. Indicates whether to normalize the distance matrices or not, default is FALSE. This is recommended if different distance types are used. More details on normalization in Normalization
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible".

Value

The returned value is a list of two elements:

DistM	The resulting distance matrix
Clust	The resulting hierarchical structure

The value has class 'HEC'.

References

Zheng L, Li T and Ding C (2014). "A Framework for Hierarchical Ensemble Clustering." *ACM Transactions on Knowledge Discovery from Data*, **9**(2), pp. 9:1–9:23.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_HEC=HierarchicalEnsembleClustering(List=L,type="data",distmeasure=
c("tanimoto","tanimoto"),normalize=c(FALSE,FALSE),method=c(NULL,NULL),
clust="agnes",linkage=c("flexible","flexible"),alpha=0.625)
```

IntClust *Integrated Clustering Methods.*

Description

The IntClust package contains several multi-source clustering methods for the integration of multiple data sets.

LabelCols *Colouring labels*

Description

Internal function of `LabelPlot`.

Usage

```
LabelCols(x, Sel1, Sel2 = NULL, col1 = NULL, col2 = NULL)
```

Arguments

`x`	The leaf of a dendrogram.
`Sel1`	The selection of objects to be colored. Default is NULL.
`Sel2`	An optional second selection to be colored. Default is NULL.
`col1`	The color for the first selection. Default is NULL.
`col2`	The color for the optional second selection. Default is NULL.

LabelPlot *Coloring specific leaves of a dendrogram*

Description

The function plots a dendrogrmam of which specific leaves are coloured.

Usage

```
LabelPlot(Data, sel1, sel2 = NULL, col1 = NULL, col2 = NULL)
```

Arguments

`Data`	The result of a method which contains the dendrogram to be colored.
`sel1`	The selection of objects to be colored. Default is NULL.
`sel2`	An optional second selection to be colored. Default is NULL.
`col1`	The color for the first selection. Default is NULL.
`col2`	The color for the optional second selection. Default is NULL.

Value

A plot of the dendrogram of which the leaves of the selection(s) are colored.

Examples

```
data(fingerprintMat)
MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

ClustF_6=cutree(MCF7_F$Clust,6)

SelF=rownames(fingerprintMat)[ClustF_6==6]
SelF

LabelPlot(Data=MCF7_F,sel1=SelF,sel2=NULL,col1='darkorchid')
```

LinkBasedClustering *Link based clustering*

Description

The LinkBasedClustering includes the ensemble clustering methods cts, srs and asrs which are voting-based consensus methods.

Usage

```
LinkBasedClustering(List, type = c("data", "dist", "clust"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), clust = "agnes", linkage = c("flexible",
  "flexible"), alpha = 0.625, nrclusters = c(7, 7), gap = FALSE,
  maxK = 15, linkBasedMethod = c("cts", "srs", "asrs"), decayfactor = 0.8,
  niter = 5, linkBasedLinkage = "ward", waitingtime = 300,
  file_number = 0, executable = FALSE)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or"clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.

method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
nrclusters	The number of clusters to divide each individual dendrogram in. Default is c(7,7) for two data sets.
gap	Logical. Whether the optimal number of clusters should be determined with the gap statistic. Defaults to FALSE.
maxK	The maximal number of clusters to investigate in the gap statistic. Default is 15.
linkBasedMethod	The method to be performed: "cts", "srs", "asrs".
decayfactor	The decay factor to be specified for the methods. Defaults to 0.8.
niter	The number of iterations. Default is 5.
linkBasedLinkage	The linkage to be used in the final clustering. Default is "ward".
waitingtime	The time in seconds to wait until the MATLAB results are generated. Defaults to 300.
file_number	The specific file number to be placed as a tag in the file generated by MATLAB. Defaults to 00.
executable	Logical. Whether the MATLAB functions are performed via an executable on the command line (TRUE, only possible for Linux systems) or by calling on MATLAB directly (FALSE). Defaults to FALSE. The files LinkBasedClusteringcts.m (cts), LinkBasedClusteringsrs.m (srs), LinkBasedClusteringasrs.m (asrs) and and MetisAlgorithm.m are present in the inst folder to be transformed in executables.

Details

(Iam-on and Garrett 2010) describe three methods for link-based clustering based on a co-association matrix: Connected-Triple Based Similarity (CTS), SimRank Based Similarity (SRS) and approximate SimRank-based similarity (ASRS). The methods compute a similarity matrix based on additional information. CTS incorporates information regarding the shared third link between two objects. SRS works based on the assumption that neighbours are similar if their neighbours are similar as well. The ASRS is introduced as a more efficient implementation of SRS. The R code calls on the MATLAB code provided by (Iam-on and Garrett 2010). The MATLAB functions are included in the inst folder and should be located in the working directory. Shell script for the executable can be found in the inst folder as well.

Value

The returned value is a list of two elements:

DistM	The resulting distance matrix
Clust	The resulting clustering

The value has class 'LinkBased'.

References

Iam-on N and Garrett S (2010). "LinkCluE: A MATLAB Package for Link-Based Cluster Ensembles." *Journal of Statistical Software*, **36**(9), pp. 1-36.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_cts=LinkBasedClustering(List=L,type="data",distmeasure=c("tanimoto", "tanimoto")
,normalize=c(FALSE,FALSE),method=c(NULL,NULL),clust="agnes",linkage = c("flexible",
"flexible"),alpha=0.625,nrclusters=c(7,7),gap = FALSE, maxK = 15,linkBasedMethod="cts",
decayfactor=0.8,niter=5,linkBasedLinkage="ward",waitingtime=300,file_number=00)

## End(Not run)
```

M_ABC *Multi-source ABC clustering*

Description

The Aggregating Bundles of Clusters (ABC, (Amaratunga et al. 2008)) was originally developed for a single gene expression data. We extend this method to incorporate multiple data sets of any source. Multi-Source ABC (M-ABC) is an iterative algorithm in which for each iteration a random sample of objects and features is taken of each data set. A clustering algorithm is run on each subset and an incidence matrix C is set up by dividing the resulting dendrogram in k clusters. After r iterations, all incidence matrices are summed and divided by number of times two objects were selected simultaneously. This similarity value is transformed into a dissimilarity measure expressing the number of times the objects are not clustered together when both are selected. The obtained matrix is used a input into a clustering algorithm.

Usage

```
M_ABC(List, transpose = TRUE, distmeasure = c("tanimoto", "tanimoto"),
  weighting = c(FALSE, FALSE), stat = "var", normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), gr = c(), bag = TRUE, numsim = 1000,
  numvar = c(100, 100), linkage = c("flexible", "flexible"),
  alpha = 0.625, NC = NULL, NC2 = NULL, mds = FALSE)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
transpose	Logical, whether the data should be transposed to have the ABC orginal format of rows being the variables and columns the samples. Defaults to TRUE.
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
weighting	Logical value indicating whether the rows should be weighted in the resampling. Default is c(FALSE,FALSE) for two data sets.
stat	The statistic to be used in weighing the rows. Currently the F-statistic, Coefficient of Variation, Double Bump statistic, and Variance are allowed. The corresponding inputs for these should be "F", "cv", "db", and "var".If the rows are to be weighed equally, any other string will do.
normalize	Logical. Indicates whether to normalize the distance matrices or not, default is FALSE. This is recommended if different distance types are used. More details on normalization in Normalization
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
gr	A prespecified grouping of the samples to be used in calculating the F-statistic if stat="F".
bag	Logical, indicating whether the columns should be bagged in each iteration. Defaults to TRUE.
numsim	The number of iterations to be used in the ABC Algorithm. Defaults to 1000.
numvar	The number of featurus to be used at each iteration to calculate the temporary clusters in the ABC Algorithm. Default is c(100,100) for two data sets.
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"
NC	Expected number of clusters in the data; passed to Wards Method in each iteration.
NC2	Expected number of clusters in the data; passed to Wards Method in the final calculation of the clusters. By default set to NC. If NC2="syl", a silhouette will be used to determine the most likely number of clusters.
mds	Logical, indicating whether the dissimilarities calculated in the ABC Algorithm should be plotted using Multi Dimensional Scaling. Defaults to FALSE

Value

The returned value is a list of two elements:

DistM	The resulting distance matrix matrix
Clust	The resulting clustering

The value has class 'Ensemble'.

References

Amaratunga D, Cabrera J and Kovtun V (2008). "Microarray learning with ABC." *Biostatistics*, **9**, pp. 128-136.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_MABC=M_ABC(List=L,transpose=TRUE,distmeasure=c("tanimoto", "tanimoto"),
weighting=c(FALSE,FALSE),stat="var",normalize=c(FALSE,FALSE),method=c(NULL,NULL),
gr=c(),bag=TRUE, numsim=1000,numvar=c(100,100),linkage=c("flexible","flexible"),
alpha=0.625,NC=7, NC2=NULL, mds=FALSE)
```

Normalization *Normalization of features*

Description

If data of different scales are being employed by the user, it is recommended to perform a normalization to make the data structures comparable.

Usage

```
Normalization(Data, method = c("Quantile", "Fisher-Yates", "Standardize",
  "Range", "Q", "q", "F", "f", "S", "s", "R", "r"))
```

Arguments

`Data`	A data matrix. It is assumed the rows are corresponding with the objects.
`method`	A method of normalization. Should be one of "Quantile","Fisher-Yates","standardize","Range" or any of the first letters of these names.

Details

The method "Quantile" refers to the Quantile-Normalization widely used in omics data. The "Fisher-Yates" normalization has a similar approach as the Quantile- Normalization but does not rely on the data, just on the number of rows present in the data matrix. The "Standardize" method refers to the `stdize` function of the **pls** package and centers and scales the data matrix. The method "Range" computes the maximum and minimum value of the matrix and determines the range. Every value is then reduced by the minimum and divided by the range of the data matrix. The latter normalization will result in values between 0 and 1.

Value

The returned value is a normalized matrix.

Examples

```
x=matrix(rnorm(100),ncol=10,nrow=10)
Norm_x=Normalization(x,method="R")
```

PathwayAnalysis *Pathway Analysis*

Description

The PathwayAnalysis function combines the functions PathwaysIter and Geneset.intersect such that only one function should be called.

Usage

```
PathwayAnalysis(List, Selection = NULL, geneExpr = NULL,
  nrclusters = NULL, method = c("limma", "MLP"), geneInfo = NULL,
  geneSetSource = "GOBP", topP = NULL, topG = NULL, GENESET = NULL,
  sign = 0.05, niter = 10, fusionsLog = TRUE, weightclust = TRUE,
  names = NULL, seperatetables = FALSE, separatepvals = FALSE)
```

Arguments

List	A list of clustering outputs or output of theDiffGenes function. The first element of the list will be used as the reference in ReorderToReference. The output of ChooseFeatures is also accepted.
Selection	If pathway analysis should be conducted for a specific selection of objects, this selection can be provided here. Selection can be of the type "character" (names of the objects) or "numeric" (the number of specific cluster). Default is NULL.
geneExpr	The gene expression matrix of the objects. The rows should correspond with the genes.
nrclusters	The number of clusters to cut the dendrogram in. Default is NULL.
method	The method to applied to look for differentially expressed genes and related pathways. For now, only the limma method is available for gene analysis and the MLP method for pathway analysis. Default is c("limma","MLP").
geneInfo	A data frame with at least the columns ENTREZID and SYMBOL. This is necessary to connect the symbolic names of the genes with their EntrezID in the correct order. The order of the gene is here not in the order of the rownames of the gene expression matrix but in the order of their significance. Default is NULL.
geneSetSource	The source for the getGeneSets function, defaults to "GOBP".
topP	Overrules sign. The number of pathways to display for each cluster. If not specified, only the significant genes are shown. Default is NULL.
topG	Overrules sign. The number of top genes to be returned in the result. If not specified, only the significant genes are shown. Default is NULL.

GENESET	Optional. Can provide own candidate gene sets. Default is NULL.
sign	The significance level to be handled. Default is 0.05.
niter	The number of times to perform pathway analysis. Default is 10.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE
weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
names	Optional. Names of the methods. Default is NULL.
seperatetables	Logical. If TRUE, a separate element is created per cluster. containing the pathways for each iteration. Default is FALSE.
separatepvals	Logical. If TRUE, the p-values of the each iteration of each pathway in the intersection is given. If FALSE, only the mean p-value is provided. Default is FALSE.

Value

The output is a list with an element per method. For each method, it is portrayed per cluster which pathways belong to the intersection over all iterations and their corresponding mean p-values.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(geneMat)
data(GeneInfo)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
names=c('FP','TP')

MCF7_PathsFandT=PathwayAnalysis(List=L, geneExpr = geneMat, nrclusters = 7, method = c("limma",
"MLP"), geneInfo = GeneInfo, geneSetSource = "GOBP", topP = NULL,
topG = NULL, GENESET = NULL, sign = 0.05,niter=2,fusionsLog = TRUE, weightclust = TRUE,
 names =names,seperatetables=FALSE,separatepvals=FALSE)

## End(Not run)
```

Pathways *Pathway analysis for multiple clustering results*

Description

A pathway analysis per cluster per method is conducted.

Usage

```
Pathways(List, Selection = NULL, geneExpr = NULL, nrclusters = NULL,
  method = c("limma", "MLP"), geneInfo = NULL, geneSetSource = "GOBP",
  topP = NULL, topG = NULL, GENESET = NULL, sign = 0.05,
  fusionsLog = TRUE, weightclust = TRUE, names = NULL)
```

Arguments

List	A list of clustering outputs or output of theDiffGenes function. The first element of the list will be used as the reference in ReorderToReference. The output of ChooseFeatures is also accepted.
Selection	If pathway analysis should be conducted for a specific selection of objects, this selection can be provided here. Selection can be of the type "character" (names of the objects) or "numeric" (the number of specific cluster). Default is NULL.
geneExpr	The gene expression matrix or ExpressionSet of the objects. The rows should correspond with the genes.
nrclusters	Optional. The number of clusters to cut the dendrogram in. The number of clusters should not be specified if the interest lies only in a specific selection of objects which is known by name. Otherwise, it is required. Default is NULL.
method	The method to applied to look for differentially expressed genes and related pathways. For now, only the limma method is available for gene analysis and the MLP method for pathway analysis. Default is c("limma","MLP").
geneInfo	A data frame with at least the columns ENTREZID and SYMBOL. This is necessary to connect the symbolic names of the genes with their EntrezID in the correct order. The order of the gene is here not in the order of the rownames of the gene expression matrix but in the order of their significance. Default is NULL.
geneSetSource	The source for the getGeneSets function, defaults to "GOBP".
topP	Overrules sign. The number of pathways to display for each cluster. If not specified, only the significant genes are shown. Default is NULL.
topG	Overrules sign. The number of top genes to be returned in the result. If not specified, only the significant genes are shown. Defaults is NULL.
GENESET	Optional. Can provide own candidate gene sets. Default is NULL.
sign	The significance level to be handled. Default is 0.05.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE

weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
names	Optional. Names of the methods. Default is NULL.

Details

After finding differently expressed genes, it can be investigated whether pathways are related to those genes. This can be done with the help of the function Pathways which makes use of the MLP function of the MLP package. Given the output of a method, the cutree function is performed which results into a specific number of clusters. For each cluster, the limma method is performed comparing this cluster to the other clusters. This to obtain the necessary p-values of the genes. These are used as the input for the MLP function to find interesting pathways. By default the candidate gene sets are determined by the AnnotateEntrezIDtoGO function. The default source will be GOBP, but this can be altered. Further, it is also possible to provide own candidate gene sets in the form of a list of pathway categories in which each component contains a vector of Entrez Gene identifiers related to that particular pathway. The default values for the minimum and maximum number of genes in a gene set for it to be considered were used. For MLP this is respectively 5 and 100. If a list of outputs of several methods is provided as data input, the cluster numbers are rearranged according to a reference method. The first method is taken as the reference and ReorderToReference is applied to get the correct ordering. When the clusters haven been re-appointed, the pathway analysis as described above is performed for each cluster of each method.

Value

The returned value is a list with an element per cluster per method. This element is again a list with the following four elements:

objects	A list with the elements LeadCpds (the objects of interest) and OrderedCpds (all objects in the order of the clustering result)
Characteristics	The found (top) characteristics of the feauture data
Genes	A list with the elements TopDE (a table with information on the top genes) and AllDE (a table with information on all genes)
Pathways	A list with the element ranked.genesets.table which is a data frame containing the genesets, their p-values and their descriptions. The second element is nr.genesets and contains the used and total number of genesets.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(geneMat)
data(GeneInfo)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
```

```
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
names=c('FP','TP')

MCF7_PathsFandT=Pathways(List=L, geneExpr = geneMat, nrclusters = 7, method = c("limma",
"MLP"), geneInfo = GeneInfo, geneSetSource = "GOBP", topP = NULL,
topG = NULL, GENESET = NULL, sign = 0.05,fusionsLog = TRUE, weightclust = TRUE,
 names =names)

## End(Not run)
```

PathwaysIter *Iterations of the pathway analysis*

Description

The MLP method to perform pathway analysis is based on resampling of the data. Therefore it is recommended to perform the pathway analysis multiple times to observe how much the results are influenced by a different resample. The function PathwaysIter performs the pathway analysis as described in Pathways a specified number of times. The input can be one data set or a list as in Pathway.2 and Pathways.

Usage

```
PathwaysIter(List, Selection = NULL, geneExpr = NULL, nrclusters = NULL,
  method = c("limma", "MLP"), geneInfo = NULL, geneSetSource = "GOBP",
  topP = NULL, topG = NULL, GENESET = NULL, sign = 0.05, niter = 10,
  fusionsLog = TRUE, weightclust = TRUE, names = NULL)
```

Arguments

List	A list of clustering outputs or output of theDiffGenes function. The first element of the list will be used as the reference in ReorderToReference. The output of ChooseFeatures is also accepted.
Selection	If pathway analysis should be conducted for a specific selection of objects, this selection can be provided here. Selection can be of the type "character" (names of the objects) or "numeric" (the number of specific cluster). Default is NULL.
geneExpr	The gene expression matrix of the objects. The rows should correspond with the genes.
nrclusters	The number of clusters to cut the dendrogram in. Default is NULL.
method	The method to applied to look for differentially expressed genes and related pathways. For now, only the limma method is available for gene analysis and the MLP method for pathway analysis. Default is c("limma","MLP").

geneInfo	A data frame with at least the columns ENTREZID and SYMBOL. This is necessary to connect the symbolic names of the genes with their EntrezID in the correct order. The order of the gene is here not in the order of the rownames of the gene expression matrix but in the order of their significance. Default is NULL.
geneSetSource	The source for the getGeneSets function ("GOBP", "GOMF","GOCC", "KEGG" or "REACTOME"). Default is "GOBP".
topP	Overrules sign. The number of pathways to display for each cluster. If not specified, only the significant genes are shown. Default is NULL.
topG	Overrules sign. The number of top genes to be returned in the result. If not specified, only the significant genes are shown. Default is NULL.
GENESET	Optional. Can provide own candidate gene sets. Default is NULL.
sign	The significance level to be handled. Default is 0.05.
niter	The number of times to perform pathway analysis. Default is 10.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE
weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
names	Optional. Names of the methods. Default is NULL.

Value

This element is again a list with the following four elements:

objects	A list with the elements LeadCpds (the objects of interest) and OrderedCpds (all objects in the order of the clustering result)
Characteristics	The found (top) characteristics of the feauture data
Genes	A list with the elements TopDE (a table with information on the top genes) and AllDE (a table with information on all genes)
Pathways	A list with the element ranked.genesets.table which is a data frame containing the genesets, their p-values and their descriptions. The second element is nr.genesets and contains the used and total number of genesets.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(geneMat)
data(GeneInfo)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
```

```
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
names=c('FP','TP')

MCF7_Paths_FandT=PathwaysIter(List=L, geneExpr = geneMat, nrclusters = 7, method =
c("limma", "MLP"), geneInfo = GeneInfo, geneSetSource = "GOBP", topP = NULL,
topG = NULL, GENESET = NULL, sign = 0.05,niter=2,fusionsLog = TRUE,
weightclust = TRUE, names =names)

## End(Not run)
```

PathwaysSelection *Pathway analysis for a selection of objects*

Description

Internal function of Pathways.

Usage

```
PathwaysSelection(List = NULL, Selection, geneExpr = NULL,
  nrclusters = NULL, method = c("limma", "MLP"), geneInfo = NULL,
  geneSetSource = "GOBP", topP = NULL, topG = NULL, GENESET = NULL,
  sign = 0.05, fusionsLog = TRUE, weightclust = TRUE, names = NULL)
```

Arguments

List	A list of clustering outputs or output of theDiffGenes function. The first element of the list will be used as the reference in ReorderToReference. The output of ChooseFeatures is also accepted.
Selection	If pathway analysis should be conducted for a specific selection of objects, this selection can be provided here. Selection can be of the type "character" (names of the objects) or "numeric" (the number of specific cluster). Default is NULL.
geneExpr	The gene expression matrix or ExpressionSet of the objects. The rows should correspond with the genes.
nrclusters	Optional. The number of clusters to cut the dendrogram in. The number of clusters should not be specified if the interest lies only in a specific selection of objects which is known by name. Otherwise, it is required. Default is NULL.
method	The method to applied to look for differentially expressed genes and related pathways. For now, only the limma method is available for gene analysis and the MLP method for pathway analysis. Default is c("limma","MLP").
geneInfo	A data frame with at least the columns ENTREZID and SYMBOL. This is necessary to connect the symbolic names of the genes with their EntrezID in the correct order. The order of the gene is here not in the order of the rownames of the gene expression matrix but in the order of their significance. Default is NULL.

geneSetSource	The source for the getGeneSets function, defaults to "GOBP".
topP	Overrules sign. The number of pathways to display for each cluster. If not specified, only the significant genes are shown. Default is NULL.
topG	Overrules sign. The number of top genes to be returned in the result. If not specified, only the significant genes are shown. Defaults is NULL.
GENESET	Optional. Can provide own candidate gene sets. Default is NULL.
sign	The significance level to be handled. Default is 0.05.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE
weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
names	Optional. Names of the methods. Default is NULL.

PlotPathways *A GO plot of a pathway analysis output.*

Description

The PlotPathways function takes an output of the PathwayAnalysis function and plots a GO graph with the help of the plotGOgraph function of the MLP package.

Usage

```
PlotPathways(Pathways, nRow = 5, main = NULL, plottype = "new",
  location = NULL)
```

Arguments

Pathways	One element of the output list returned by PathwayAnalysis or Geneset.intersect.
nRow	Number of GO IDs for which to produce the plot. Default is 5.
main	Title of the plot. Default is NULL.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Value

The output is a GO graph.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(geneMat)
data(GeneInfo)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
names=c('FP','TP')

MCF7_PathsFandT=PathwayAnalysis(List=L, geneExpr = geneMat, nrclusters = 7, method = c("limma",
"MLP"), geneInfo = GeneInfo, geneSetSource = "GOBP", topP = NULL,
topG = NULL, GENESET = NULL, sign = 0.05,niter=2,fusionsLog = TRUE, weightclust = TRUE,
 names =names,seperatetables=FALSE,separatepvals=FALSE)

PlotPathways(MCF7_PathsFandT$FP$"Cluster 1"$Pathways,nRow=5,main=NULL)

## End(Not run)
```

`PreparePathway` *Preparing a data set for pathway analysis*

Description

The functions for pathway analysis in this package can also work on results of the integrated data functions. However, a differential gene expression needs to be conducted to perform pathway analysis. The function `PreparePathway` checks if the necessary elements are present in the data structures and if not, the elements such as p-values are created. It is an internal function to all pathway analysis functions but can be used separately as well.

Usage

```
PreparePathway(Object, geneExpr, topG, sign)
```

Arguments

`Object`	A list with at least an element with the name "objects" such that the function knows which objects to test for differential gene expression. If the elements "Genes" and "pvalsgenes" are present as well, these will be collected and the gene expression is not analyzed.
`geneExpr`	The gene expression matrix or ExpressionSet of the objects. The rows should correspond with the genes.

topG	Overrules sign. The number of top genes to be returned in the result. If not specified, only the significant genes are shown. Default is NULL.
sign	The significance level to be handled. Default is 0.05.

Value

The returned value is a list with three elements:

pvalsgenses	This is a list with that contains a vector of raw p-values for every group of tested objects.
objects	This is a list with that contains another list per group of tested objects. Every list contains the lead objects and the ordered objects.
Genes	This is a list with that contains contains another list per group of tested objects. Every list contains two data frames, one with information on the top genes and one with information on all genes.

Examples

```
data(fingerprintMat)
data(geneMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)


L1=list(MCF7_F)

Comps1=FindCluster(L1,nrclusters=7,select=c(1,1))
Comps2=FindCluster(L1,nrclusters=7,select=c(1,2))
Comps3=FindCluster(L1,nrclusters=7,select=c(1,3))


L2=list()

L2$'Cluster 1'$objects$LeadCpds=Comps1
L2$'Cluster 2'$objects$LeadCpds=Comps2
L2$'Cluster 3'$objects$LeadCpds=Comps2

MCF7_PreparePaths=PreparePathway(Object=L2,geneExpr=geneMat,topG=NULL,sign=0.05)
str(MCF7_PreparePaths)
```

ProfilePlot *Plotting gene profiles*

Description

In ProfilePlot, the gene profiles of the significant genes for a specific cluster are shown on 1 plot. Therefore, each gene is normalized by subtracting its the mean.

Usage

```
ProfilePlot(Genes, Comps, geneExpr = NULL, raw = FALSE, orderLab = NULL,
  colorLab = NULL, nrclusters = NULL, cols = NULL, addLegend = TRUE,
  margins = c(8.1, 4.1, 1.1, 6.5), extra = 5, plottype = "new",
  location = NULL)
```

Arguments

Genes	The genes to be plotted.
Comps	The objects to be plotted or to be separated from the other objects.
geneExpr	The gene expression matrix or ExpressionSet of the objects.
raw	Logical. Should raw p-values be plotted? Default is FALSE.
orderLab	Optional. If the objects are to set in a specific order of a specific method. Default is NULL.
colorLab	The clustering result that determines the color of the labels of the objects in the plot. Default is NULL.
nrclusters	Optional. The number of clusters to cut the dendrogram in.
cols	Optional. The color to use for the objects in Clusters for each method.
addLegend	Optional. Whether a legend of the colors should be added to the plot.
margins	Optional. Margins to be used for the plot. Default is margins=c(8.1,4.1,1.1,6.5).
extra	The space between the plot and the legend. Default is 5.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Value

A plot which contains multiple gene profiles. A distinction is made between the values for the objects in Comps and the others.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(geneMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
```

```
L=list(MCF7_F,MCF7_T)
names=c('FP','TP')

MCF7_FT_DE = DiffGenes(List=L,geneExpr=geneMat,nrclusters=7,method="limma",sign=0.05,topG=10,
fusionsLog=TRUE,weightclust=TRUE)

Comps=SharedComps(list(MCF7_FT_DE$`Method 1`$"Cluster 1",MCF7_FT_DE$`Method 2`$"Cluster 1"))[[1]]

MCF7_SharedGenes=FindGenes(dataLimma=MCF7_FT_DE,names=c("FP","TP"))

Genes=names(MCF7_SharedGenes[[1]])[-c(2,4,5)]

colscl=ColorPalette(colors=c("red","green","purple","brown","blue","orange"),ncols=9)

ProfilePlot(Genes=Genes,Comps=Comps,geneExpr=geneMat,raw=FALSE,orderLab=MCF7_F,
colorLab=NULL,nrclusters=7,cols=colscl,addLegend=TRUE,margins=c(16.1,6.1,1.1,13.5),
extra=4,plottype="sweave",location=NULL)

## End(Not run)
```

`ReorderToReference` *Order the outputs of the clustering methods against a reference*

Description

When multiple methods are performed on a data set, it is interesting to compare their results. However, a comparison is not easily done since different methods leads to a different ordering of the objects. The `ReorderToReference` rearranges the cluster to a reference method.

Usage

```
ReorderToReference(List, nrclusters = NULL, fusionsLog = FALSE,
  weightclust = FALSE, names = NULL)
```

Arguments

`List`	A list of clustering outputs to be compared. The first element of the list will be used as the reference.
`nrclusters`	The number of clusters to cut the dendrogram in. Default is NULL.
`fusionsLog`	Logical. Indicator for the fusion of clusters. Default is FALSE.
`weightclust`	Logical. To be handed to `ReorderToReference`: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
`names`	Optional. A character vector with the names of the methods.

Details

It is interesting to compare the results of the methods described in the methodology. All methods result in a dendrogram which is cut into a specific number of clusters with the `cutree` function. This results in an numbering of cluster based on the ordering of the names in the data and not on the order in which they are grouped into clusters. However, different methods lead to different clusters and it is possible that cluster 1 of one method will not be the cluster that has the most in common with cluster 1 of another method. This makes comparisons rather difficult. Therefore the ReorderToReference function was written which takes one method as a reference and rearranges the cluster numbers of the other methods to this reference such that clusters are appointed to that cluster they have the most in common with. The result of this function is a matrix of which the columns are in the order of the clustering of the objects of the referenced method and the rows represent the methods. Each cell contains the number of the cluster the compound is in for that method compared to the method used as a reference. This function is applied in the functions `SimilarityMeasure`, `DiffGenes`, `Pathways` and `ComparePlot`. It is a possibility that 2 or more clusters are fused together compared to the reference method. If this is true, the function will alert the user and will ask to put the parameter fusionsLog to true. Since `ReorderToReference` is often used as an internal function, also for visualization, it will print out how many more colors should be specified for those clusters that did not find a suitable match. This can be due to fusion or complete segregation of its objects into other clusters.

Value

A matrix of which the cells indicate to what cluster the objects belong to according to the rearranged methods.

Note

The `ReorderToReference` function was optimized for the situations presented by the data sets at hand. It is noted that the function might fail in a particular situation which results in a infinite loop.

Examples

```
data(fingerprintMat)
data(targetMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_ADC=ADC(list(fingerprintMat,targetMat),distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible")

L=list(MCF7_F,MCF7_ADC,MCF7_T)
names=c("FP","ADC","TP")

MCF7_Matrix=ReorderToReference(List=L,nrclusters = 7, fusionsLog = FALSE, weightclust =
FALSE, names = names)
```

SelectnrClusters *Determines an optimal number of clusters based on silhouette widths*

Description

The function SelectnrClusters determines an optimal optimal number of clusters based by calculating silhouettes widths for a sequence of clusters. See "Details" for a more elaborate description.

If the object provided in List are data or distance matrices clustering around medoids is performed with the pam function of the **cluster** package. Of the obtained pam objects, average silhouette widths are retrieved. A silhouette width represents how well an object lies in its current cluster. Values around one are an indication of an appropriate clustering while values around zero show that the object might as well lie in the neighbouring cluster. The average silhouette width is a measure of how tightly grouped the data is. This is performed for every number of cluster for every object provided in List. Then the average is taken for every number of clusters over the provided objects. This results in one average value per number of clusters. The number width the maximal average silhouette width is chosen as the optimal number of clusters.

Usage

```
SelectnrClusters(List, type = c("data", "dist", "pam"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), nrclusters = seq(5, 25, 1), names = NULL,
  StopRange = FALSE, plottype = "new", location = NULL)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
nrclusters	A sequence of numbers of clusters to cut the dendrogram in. Default is a sequence of 5 to 25.
names	The labels to give to the elements in List. Default is NULL.

StopRange — Logical. Indicates whether the distance matrices with values not between zero and one should be standardized to have so. If FALSE the range normalization is performed. See Normalization. If TRUE, the distance matrices are not changed. This is recommended if different types of data are used such that these are comparable. Default is FALSE.

plottype — Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".

location — If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Value

A plots are made showing the average silhouette widths of the provided objects for each number of clusters. Further, a list with two elements is returned:

Silhouette_Widths — A data frame with the silhouette widths for each object and the average silhouette widths per number of clusters

Optimal_Nr_of_CLusters — The determined optimal number of cluster

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)

L=list(fingerprintMat,targetMat)

NrClusters=SelectnrClusters(List=L,type="data",distmeasure=c("tanimoto",
"tanimoto"),nrclusters=seq(5,10),normalize=c(FALSE,FALSE),method=c(NULL,NULL),
names=c("FP","TP"),StopRange=FALSE,plottype="new",location=NULL)

NrClusters

## End(Not run)
```

SharedComps *Intersection of clusters across multiple methods*

Description

The SharedComps function is an easy way to select the objects that are shared over clusters of different methods.

Usage

```
SharedComps(List, nrclusters = NULL, fusionsLog = FALSE,
  weightclust = FALSE, names = NULL)
```

Arguments

List	A list of clustering outputs or the output of the DiffGenes function. The first element of the list will be used as a reference in ReorderToReference.
nrclusters	If List is the output several clustering methods, it has to be provided in how many clusters to cut the dendrograms in. Default is NULL.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is FALSE
weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is FALSE.
names	Names of the methods or clusters. Default is NULL.

Value

A vector containing the shared objects of all listed elements.

Examples

```
data(fingerprintMat)
data(targetMat)
data(geneMat)
data(GeneInfo)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
names=c('FP','TP')

Comps=SharedComps(List=L,nrclusters=7,fusionsLog=FALSE,weightclust=FALSE,names=names)
```

SharedGenesPathsFeat *Intersection of genes and pathways over multiple methods*

Description

It is interesting to investigate exactly which and how many differently expressed genes, pathways and characteristics are shared by the clusters over the different methods. The function SharedGenesPathsFeat will provide this information. Given the outputs of the DiffGenes, the Geneset.intersect function and/or CharacteristicFeatures, it investigates how many genes, pathways and/or characteristics are expressed by each cluster per method, how many of these are shared over the methods and which ones are shared including their respective p-values of each method and a mean p-value. This is very handy to look into the shared genes and pathways of clusters that share many objects but also of those that only share only a few. Further, the result also includes the number of objects per cluster per method and how many of these are shared over the methods. The input can also be focused for a specific selection of objects or a specific cluster.

Usage

```
SharedGenesPathsFeat(DataLimma = NULL, DataMLP = NULL, DataFeat = NULL,
  names = NULL, Selection = FALSE)
```

Arguments

DataLimma	Optional. The output of a DiffGenes function. Default is NULL.
DataMLP	Optional. The output of Geneset.intersect function. Default is NULL.
DataFeat	Optional. The output of CharacteristicFeatures function. Default is NULL.
names	Optional. Names of the methods or "Selection" if it only considers a selection of objects. Default is NULL.
Selection	Logical. Do the results concern only a selection of objects or a specific cluster? If yes, then Selection should be put to TRUE. Otherwise all objects and clusters are considered.Default is FALSE.

Value

The result of the SharedGenesPathsFeat function is a list with two elements. The first element Table is a table indicating how many genes, pathways and/or characteristics were found to be differentially expressed and how many of these are shared. The table also contains the number of objects shared between the clusters of the different methods. The second element Which is another list with a component per cluster. Each component consists of four vectors: SharedComps indicating which objects were shared across the methods, SharedGenes represents the shared genes, SharedPaths shows the shared pathways and SharedFeat the shared features.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(geneMat)
data(GeneInfo)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
```

```
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
names=c('FP','TP')

MCF7_Paths_FandT=PathwaysIter(List=L, geneExpr=geneMat, nrclusters=7, method=
c("limma", "MLP"), geneInfo=GeneInfo, geneSetSource="GOBP", topP=NULL,
topG=NULL, GENESET=NULL, sign=0.05,niter=2,fusionsLog=TRUE,
weightclust=TRUE, names =names)

MCF7_Paths_intersection=Geneset.intersect(MCF7_Paths_FandT,0.05,names=names,
seperatetables=FALSE,separatepvals=FALSE)

MCF7_DiffGenes_FandT10=DiffGenes(list(MCF7_F,MCF7_T),Selection=NULL,geneExpr=geneMat,
nrclusters=7,method="limma",sign=0.05,top=10,fusionsLog=TRUE,weightclust=TRUE,names=NULL)

MCF7_Char=CharacteristicFeatures(list(MCF7_F,MCF7_T),Selection=NULL,binData=
list(fingerprintMat,targetMat),datanames=c("FP","TP"),nrclusters=7,top=NULL,
sign=0.05,fusionsLog=TRUE,weightclust=TRUE,names=c("FP","TP"))

MCF7_Shared=SharedGenesPathsFeat(DataLimma=MCF7_DiffGenes_FandT10,
DataMLP=MCF7_Paths_intersection,DataFeat=MCF7_Char)

str(MCF7_Shared)

## End(Not run)
```

`SharedSelection` *Intersection of genes and pathways over multiple methods for a selection of objects.*

Description

Internal function of `SharedGenesPathsFeat`.

Usage

```
SharedSelection(DataLimma = NULL, DataMLP = NULL, DataFeat = NULL,
  names = NULL)
```

Arguments

`DataLimma`	Optional. The output of a `DiffGenes` function. Default is NULL.
`DataMLP`	Optional. The output of `Geneset.intersect` function. Default is NULL.
`DataFeat`	Optional. The output of `CharacteristicFeatures` function. Default is NULL.
`names`	Optional. Names of the methods or "Selection" if it only considers a selection of objects. Default is NULL.

SharedSelectionLimma *Intersection of genes over multiple methods for a selection of objects.*

Description

Internal function of SharedGenesPathsFeat.

Usage

```
SharedSelectionLimma(DataLimma = NULL, names = NULL)
```

Arguments

DataLimma	Optional. The output of a DiffGenes function. Default is NULL.
names	Optional. Names of the methods or "Selection" if it only considers a selection of objects. Default is NULL.

SharedSelectionMLP *Intersection of pathways over multiple methods for a selection of objects.*

Description

Internal function of SharedGenesPathsFeat.

Usage

```
SharedSelectionMLP(DataMLP = NULL, names = NULL)
```

Arguments

DataMLP	Optional. The output of Geneset.intersect function. Default is NULL.
names	Optional. Names of the methods or "Selection" if it only considers a selection of objects. Default is NULL.

`SimilarityHeatmap` *A heatmap of similarity values between objects*

Description

The function `SimilarityHeatmap` plots the similarity values between objects. The darker the shade, the more similar objects are. The option is available to set a cutoff value to highlight the most similar objects.

Usage

```
SimilarityHeatmap(Data, type = c("data", "clust", "sim", "dist"),
  distmeasure = "tanimoto", normalize = FALSE, method = NULL,
  linkage = "flexible", cutoff = NULL, percentile = FALSE,
  plottype = "new", location = NULL)
```

Arguments

`Data`	The data of which a heatmap should be drawn.
`type`	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" ,"sim" or "clusters".
`distmeasure`	The distance measure. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to "tanimoto".
`normalize`	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in `Normalization`.
`method`	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is NULL.
`linkage`	Choice of inter group dissimilarity (character). Defaults to "flexible".
`cutoff`	Optional. If a cutoff value is specified, all values lower are put to zero while all other values are kept. This helps to highlight the most similar objects. Default is NULL.
`percentile`	Logical. The cutoff value can be a percentile. If one want the cutoff value to be the 90th percentile of the data, one should specify cutoff = 0.90 and percentile = TRUE. Default is FALSE.
`plottype`	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document, i.e. no new device is opened and the plot appears in the current device or document. Default is "new".
`location`	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Details

If data is of type "clust", the distance matrix is extracted from the result and transformed to a similarity matrix. Possibly a range normalization is performed. If data is of type "dist", it is also transformed to a similarity matrix and cluster is performed on the distances. If data is of type "sim", the data is tranformed to a distance matrix on which clustering is performed. Once the similarity mattrix is obtained, the cutoff value is applied and a heatmap is drawn. If no cutoff value is desired, one can leave the default NULL specification.

Value

A heatmap with the names of the objects on the right and bottom and a dendrogram of the clustering at the left and top.

Examples

```
## Not run:
data(fingerprintMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55)

SimilarityHeatmap(Data=MCF7_F,type="clust",cutoff=0.90,percentile=TRUE)
SimilarityHeatmap(Data=MCF7_F,type="clust",cutoff=0.75,percentile=FALSE)


## End(Not run)
```

SimilarityMeasure *A measure of similarity for the outputs of the different methods*

Description

The function `SimilarityMeasure` computes the similarity of the methods. Given a list of outputs as input, the first element will be seen as the reference. Function `MatrixFunction` is called upon and the cluster numbers are rearranged according to the reference. Per method, `SimilarityMeasure` investigates which objects have the same cluster number in reference and said method. This number is divided by the total number of objects and used as a similarity measure.

Usage

```
SimilarityMeasure(List, nrclusters = NULL, fusionsLog = TRUE,
  weightclust = TRUE, names = NULL)
```

Arguments

List	A list of clustering outputs to be compared. The first element of the list will be used as the reference in ReorderToReference.
nrclusters	The number of clusters to cut the dendrogram in. Default is NULL.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE
weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
names	Optional. Names of the methods.

Value

A vector of similarity measures, one for each method given as input.

Examples

```
data(fingerprintMat)
data(targetMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
names=c("FP","TP")

MCF7_SimFandT=SimilarityMeasure(List=L,nrclusters=7,fusionsLog=TRUE,weightclust=TRUE,
names=names)
```

SNF *Similarity network fusion*

Description

Similarity Network Fusion (SNF) is a similarity-based multi-source clustering technique. SNF consists of two steps. In the initial step a similarity network is set up for each data matrix. The network is the visualization of the similarity matrix as a weighted graph with the objects as vertices and the pairwise similarities as weights on the edges. In the network-fusion step, each network is iteratively updated with information of the other network which results in more alike networks every time. This eventually converges to a single network.

Usage

```
SNF(List, type = c("data", "dist", "clusters"), distmeasure = c("tanimoto",
  "tanimoto"), normalize = c(FALSE, FALSE), method = c(NULL, NULL),
  StopRange = FALSE, NN = 20, mu = 0.5, T = 20, clust = "agnes",
  linkage = "ward", alpha = 0.625)
```

Arguments

List	A list of data matrices of the same type. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
StopRange	Logical. Indicates whether the distance matrices with values not between zero and one should be standardized to have so. If FALSE the range normalization is performed. See Normalization. If TRUE, the distance matrices are not changed. This is recommended if different types of data are used such that these are comparable. Default is FALSE.
NN	The number of neighbours to be used in the procedure. Defaults to 20.
mu	The parameter epsilon. The value is recommended to be between 0.3 and 0.8. Defaults to 0.5.
T	The number of iterations.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for the final clustering. Defaults to "ward".
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"

Details

If r is specified and nrclusters is a fixed number, only a random sampling of the features will be performed for the t iterations (ADECa). If r is NULL and the nrclusters is a sequence, the clustering is performedon all features and the dendrogam is divided into clusters for the values of nrclusters (ADECb). If both r is specified and nrclusters is a sequence, the combination is performed (ADECc). After every iteration, either be random sampling, multiple divisions of the dendrogram or both, an incidence matrix is set up. All incidence matrices are summed and represent the distance matrix on which a final clustering is performed.

Value

The returned value is a list with the following three elements.

FusedM	The fused similarity matrix
DistM	The distance matrix computed by subtracting FusedM from one
Clust	The resulting clustering

The value has class 'SNF'.

References

Wang B, Mezlini MA, Demir F, Fiume M, Tu Z, Brudno M, Haibe-Kains B and Goldenberg A (2014). "Similarity Network Fusion for Aggregrating Data Types on a Genomic Scale." *Nature*, **11**(3), pp. 333-337.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)
MCF7_SNF=SNF(List=L,type="data",distmeasure=c("tanimoto","tanimoto"),normalize=
c(FALSE,FALSE),method=c(NULL,NULL),StopRange=FALSE,NN=10,mu=0.5,T=20,clust="agnes",
linkage="ward",alpha=0.625)
```

---

targetMat *Target prediction data*

Description

A binary data matrix that contains 477 target predictions for a set of 56 compounds.

Usage

```
data(targetMat)
```

Format

An object of class "matrix".

Examples

```
data(targetMat)
```

TrackCluster *Follow a cluster over multiple methods*

Description

It is often desired to track a specific selection of object over the different methods and/or weights. This can be done with the ClusterDistribution. For every method, it is tracked where the objects of the selections are situated.

Usage

```
TrackCluster(List, Selection, nrclusters = NULL, followMaxComps = FALSE,
  followClust = TRUE, fusionsLog = TRUE, weightclust = TRUE,
  names = NULL, selectionPlot = FALSE, table = FALSE,
  completeSelectionPlot = FALSE, ClusterPlot = FALSE, cols = NULL,
  legendposx = 0.5, legendposy = 2.4, plottype = "sweave",
  location = NULL)
```

Arguments

List	A list of the clustering outputs.The first element of the list will be used as the reference in ReorderToReference.
Selection	The selection of objects to follow or a specific cluster number. Default is NULL.
nrclusters	The number of clusters to cut the dendrogram in. Default is NULL.
followMaxComps	Logical for plot. Whether to follow the maximum of objects. Default is FALSE.
followClust	Logical for plot. Whether to follow the specific cluster. Default is TRUE.
fusionsLog	Logical. To be handed to ReorderToReference: indicator for the fusion of clusters. Default is TRUE
weightclust	Logical. To be handed to ReorderToReference: to be used for the outputs of CEC, WeightedClust or WeightedSimClust. If TRUE, only the result of the Clust element is considered. Default is TRUE.
names	Optional. Names of the methods. Default is NULL.
selectionPlot	Logical. Should a plot be produced. Depending on followMaxComps and followClust it focuses on the maximum of objects or a cluster. It will not be indicated to which cluster objects moved. Default is FALSE.
table	Logical. Should a table with the objects per method and the shared objects be produced? Default is FALSE.
completeSelectionPlot	Logical. Should the complete distribution of the selection be plotted? This implies that it will be indicated to which cluster objects will move. Default is FALSE.
ClusterPlot	Logical. Plot of specific cluster. Default is FALSE.
cols	The colors used for the different clusters. Default is NULL.

legendposx	The x-coordinate of the legend on all plots. Default is 0.5.
legendposy	The y-coordinate of the legend on all plots. Default is 2.4.
plottype	Should be one of "pdf","new" or "sweave". If "pdf", a location should be provided in "location" and the figure is saved there. If "new" a new graphic device is opened and if "sweave", the figure is made compatible to appear in a sweave or knitr document. Default is "new".
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there. Default is NULL.

Details

The result is provided with extra information as which objects of the original selection can be found in this cluster and which are extra. Further, plots of the distribution of the objects can be produced. One plot follows the complete distribution of the cluster while another one focuses on either the maximum number of objects or a specific cluster, whatever is specified. It are the number of objects that are plotted and the first element indicated the number of objects in the selection. A table can be produced as well, that separates the objects that are shared over all methods from those extra in the original selection and extra for the other methods. The `ReorderToReference` is applied to make sure that the clusters are comparable over the methods.

The function is experimental and might not work in specific cases. Please let us know such that we can improve its functionality.

Value

The returned value is a list with an element for every method. This element is another list with the following elements:

Selection	The selection of objects to follow
nr.clusters	the number of clusters the selection is divided over
nr.min.max.together	the minimum and maximum number of objects found together
perc.min.max.together	minimum and maximum percentage of objects found together
AllClusters	A list with an element per cluster that contains at least one of the objects in Selection. The list contains the cluster number, the complete cluster, the objects that originally could be found in this cluster and which object were joined extra to it.

Depending on whether followMaxComps or followClust is specified, the cluster of interest is mentioned separately as well for easy access. If the option was specified to create a table, this can be found under the Table element. Each plot that was specified to be created is plotted in a new window in the graphics console.

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
```

```
data(Colors1)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="flexible",gap=FALSE,maxK=55,StopRange=FALSE)

L=list(MCF7_F,MCF7_T)
names=c("FP","TP")

Comps=FindCluster(List=L,nrclusters=7,select=c(1,4))
Comps

CompsFPAll=TrackCluster(List=L,Selection=Comps,nrclusters=7,followMaxComps=TRUE,
followClust=FALSE,fusionsLog=TRUE,weightclust=TRUE,names=names,selectionPlot=TRUE,
table=TRUE,completeSelectionPlot=TRUE,cols=Colors1,plottype="new",location=NULL)

## End(Not run)
```

`WeightedClust` *Weighted clustering*

Description

Weighted Clustering (Weighted) is a similarity-based multi-source clustering technique. Weighted clustering is performed with the function `WeightedClust`. Given a list of the data matrices, a dissimilarity matrix is computed of each with the provided distance measures. These matrices are then combined resulting in a weighted dissimilarity matrix. Hierarchical clustering is performed on this weighted combination with the agnes function and the ward link

Usage

```
WeightedClust(List, type = c("data", "dist", "clusters"),
  distmeasure = c("tanimoto", "tanimoto"), normalize = c(FALSE, FALSE),
  method = c(NULL, NULL), StopRange = FALSE, weight = seq(1, 0, -0.1),
  weightclust = 0.5, clust = "agnes", linkage = "ward", alpha = 0.625)
```

Arguments

`List`	A list of data matrices. It is assumed the rows are corresponding with the objects.
`type`	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or "clusters".
`distmeasure`	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").

normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.
StopRange	Logical. Indicates whether the distance matrices with values not between zero and one should be standardized to have values between zero and one. If FALSE the range normalization is performed. See Normalization. If TRUE, the distance matrices are not changed. This is recommended if different types of data are used such that these are comparable. Default is FALSE.
weight	Optional. A list of different weight combinations for the data sets in List. If NULL, the weights are determined to be equal for each data set. It is further possible to fix weights for some data matrices and to let it vary randomly for the remaining data sets. Defaults to seq(1,0,-0.1). An example is provided in the details.
weightclust	A weight for which the result will be put aside of the other results. This was done for comparative reason and easy access. Defaults to 0.5 (two data sets)
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for the final clustering. Defaults to "ward".
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible".

Details

The weight combinations should be provided as elements in a list. For three data matrices an example could be: weights=list(c(0.5,0.2,0.3),c(0.1,0.5,0.4)). To provide a fixed weight for some data sets and let it vary randomly for others, the element "x" indicates a free parameter. An example is weights=list(c(0.7,"x","x")). The weight 0.7 is now fixed for the first data matrix while the remaining 0.3 weight will be divided over the other two data sets. This implies that every combination of the sequence from 0 to 0.3 with steps of 0.1 will be reported and clustering will be performed for each.

Value

The returned value is a list of four elements:

DistM	A list with the distance matrix for each data structure
WeightedDist	A list with the weighted distance matrices
Results	The hierarchical clustering result for each element in WeightedDist
Clust	The result for the weight specified in Clustweight

The value has class 'Weighted'.

References

Perualila-Tan N, Shkedy Z, Talloen W, Goehlmann HWH, Consortium Q, Van Moerbeke M and Kasim A (2016). "Weighted-Similarity Based Clustering of Chemical Structure and Bioactivity Data in Early Drug Discovery." *Journal of Bioinformatics and Computational Biology*, **14**(4), pp. 1650018.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_Weighted=WeightedClust(List=L,type="data",distmeasure=c("tanimoto","tanimoto"),
normalize=c(FALSE,FALSE),method=c(NULL,NULL),StopRange=FALSE,weight=seq(1,0,-0.1),
weightclust=0.5,clust="agnes",linkage="ward",alpha=0.625)
```

WonM *Weighting on membership*

Description

Weighting on Membership (WonM) is similar to CEC as the dendrograms are divided into clusters for a range of values for the number of clustes. However, instead of weighting the sum of the incidences matrices, the final matrix for clustering is the normal sum of all incidence matrices.

Usage

```
WonM(List, type = c("data", "dist", "clusters"), distmeasure = c("tanimoto",
  "tanimoto"), normalize = c(FALSE, FALSE), method = c(NULL, NULL),
  nrclusters = seq(5, 25, 1), clust = "agnes", linkage = c("flexible",
  "flexible"), alpha = 0.625)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
type	indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained from the data. If type="dist" the calculation of the distance matrices is skipped and if type="clusters" the single source clustering is skipped. Type should be one of "data", "dist" or"clusters".
distmeasure	A vector of the distance measures to be used on each data matrix. Should be one of "tanimoto", "euclidean", "jaccard", "hamming". Defaults to c("tanimoto","tanimoto").
normalize	Logical. Indicates whether to normalize the distance matrices or not, defaults to c(FALSE, FALSE) for two data sets. This is recommended if different distance types are used. More details on normalization in Normalization.
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names. Default is c(NULL,NULL) for two data sets.

nrclusters	A sequence of numbers of clusters to cut the dendrogram in. Defaults is a sequence of 5 to 25.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	Choice of inter group dissimilarity (character) for each data set. Defaults to c("flexible", "flexible") for two data sets.
alpha	The parameter alpha to be used in the "flexible" linkage of the agnes function. Defaults to 0.625 and is only used if the linkage is set to "flexible"

Value

The returned value is a list of two elements:

DistM	The resulting incidence matrix
Clust	The resulting clusters

The value has class 'WonM'.

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)

MCF7_WonM=WonM(List=L,type="data",distmeasure=c("tanimoto", "tanimoto"),
normalize=c(FALSE,FALSE),method=c(NULL,NULL),nrclusters=seq(5,25,1),
clust="agnes",linkage=c("flexible","flexible"),alpha=0.625)
```

Index