Package 'IntClust'

March 31, 2016

Type Package

Title Integrated Data Analysis via Clustering

Version 0.0.2

Date 2016-03-31

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

Imports ade4,a4Core, Biobase, cluster, plotrix, plyr, gplots, gridExtra, limma, prodlim, gtools,e1071,pls,stats,utils,graphics,FactoMineR,analogue,lsa, SNFtool,grDevices,ggplot2

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

NeedsCompilation no

Depends R (>= 2.10)

Repository CRAN

Date/Publication 2016-03-31 17:38:30

R topics documented:

`IntClust-package` *Integrated Data Analysis*

Description

The package contains several integrative data methods in which information of objects from different data sources can be combined. 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.

Details

Package:	IntClust
Type:	Package
Version:	0.0.18
Date:	2014-10-30
License:	GPL-3

Author(s)

Marijke Van Moerbeke

Maintainer: Marijke Van Moerbeke <marijkevanmoerbeke@uhasselt.be>

ADC *Aggregated Data Clustering*

Description

In order to perform aggregated data clustering, the ADClust function was written. The data matrices are aggregated into one and hierarchical clustering is performed.

Usage

```
ADC(List, distmeasure = "tanimoto",normalize=FALSE,method=NULL,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.
distmeasure	Choice of metric for the dissimilarity matrix (character). Should be one of "tanimoto", "euclidean", "jaccard", "hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. 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.
clust	Choice of clustering function (character). Defaults to "agnes".
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"

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.

Note

For now, only hierarchical clustering with the agnes function is implemented.

Author(s)

Marijke Van Moerbeke

References

FODEH, J. S., BRANDT, C., LUONG, B. T., HADDAD, A., SCHULTZ, M., MURPHY, T., KRAUTHAMMER, M. (2013). Complementary Ensemble Clustering of Biomedical Data. J Biomed Inform. 46(3) pp.436-443.

Examples

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

ADEC *Aggregated Data Ensemble Clustering*

Description

Function ADEC performs which the functions ADECa, ADECb and ADECc is specified by the user.

Usage

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

Arguments

List	A list of data matrices of the same type. It is assumed the rows are corresponding with the objects.
distmeasure	The distance measure to be used on the fused data matrix (character). Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. 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.
t	The number of iterations.

r	Optional. The number of features to take for the random sample.
nrclusters	The number of clusters to cut the dendrogram in. If a sequence is specified either ADECb or ADECc is performed. A fixed number of clusters defaults to ADECa
clust	Choice of clustering function (character). Defaults to "agnes".
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"
ResampleFeatures	Logical. Whether the features should be resamples. If TRUE, either ADECa or ADECc is performed.

Details

See the details of ADECa, ADECb and ADEDc for more information.

Value

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

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

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

Note

For now, only hierarchical clustering with the agnes function implemented.

Author(s)

Marijke Van Moerbeke

References

FODEH, J. S., BRANDT, C., LUONG, B. T., HADDAD, A., SCHULTZ, M., MURPHY, T., KRAUTHAMMER, M. (2013). Complementary Ensemble Clustering of Biomedical Data. J Biomed Inform. 46(3) pp.436-443.

See Also

ADECa, ADECb,ADECc

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)
MCF7_ADECa=ADEC(L,distmeasure="tanimoto",normalize=FALSE,method=NULL,t=25,r=NULL,
nrclusters=7,clust="agnes",linkage="ward",alpha=0.625,ResampleFeatures=TRUE)
```

ADECa *Aggregated Data Ensemble Clustering - version a*

Description

Function ADECa performs aggregated data ensemble clustering in which in every iteration the number of random samples taken is randomly set between m/2 and m-1 with m the total number of features. The number of features to sample can also be prespecified by the user.

Usage

```
ADECa(List, distmeasure = "tanimoto",normalize=FALSE,method=NULL, t = 10,
r = NULL, nrclusters = NULL, 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.
distmeasure	The distance measure to be used on the fused data matrix (character). Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. 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.
t	The number of iterations.
r	Optional. The number of features to take for the random sample.
nrclusters	The number of clusters to cut the dendrogram in.
clust	Choice of clustering function (character). Defaults to "agnes".
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"

Details

ADECa starts with the merging of the data matrices into one larger data matrix. Then, ensemble clustering is performed on the fused data. This comes down to repeatedly applying hierarchical clustering. A random sample of features is taken in each application. More information can be found in Fodeh et al. (2013).

Value

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

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

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

Note

For now, only hierarchical clustering with the agnes function is implemented.

Author(s)

Marijke Van Moerbeke

References

FODEH, J. S., BRANDT, C., LUONG, B. T., HADDAD, A., SCHULTZ, M., MURPHY, T., KRAUTHAMMER, M. (2013). Complementary Ensemble Clustering of Biomedical Data. J Biomed Inform. 46(3) pp.436-443.

See Also

ADEC,ADECb,ADECc

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)
MCF7_ADECa=ADECa(L,distmeasure="tanimoto",normalize=FALSE,method=NULL,t=25,r=NULL,
nrclusters=7,clust="agnes",linkage="ward",alpha=0.625)
```

ADECb *Aggregated Data Ensemble Clustering - version b*

Description

Function ADECb performs aggregated data ensemble clustering in which in every iteration the total number of features are used in the clustering procedure. However, the function is capable of cutting the resulting dendrogram several times, each time into a different number of cluster.

Usage

```
ADECb(List, distmeasure = "tanimoto",normalize=FALSE,method=NULL,
nrclusters = seq(5, 25, 1),  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.
distmeasure	The distance measure to be used on the fused data matrix (character). Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. 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.
nrclusters	A sequence of numbers of clusters to cut the dendrogram in.
clust	Choice of clustering function (character). Defaults to "agnes".
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"

Details

ADECb starts with the merging of the data matrices into one larger data matrix. Then, ensemble clustering is performed on the fused data. This comes down to repeatedly applying hierarchical clustering. All features will be used in every iteration. Variation is inserted by not splitting the dendrogram a single time into one specific number of clusters but multiple times and for a range of numbers of clusters. More information can be found in Fodeh et al. (2013).

Value

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

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

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

Note

For now, only hierarchical clustering with the agnes function is implemented.

Author(s)

Marijke Van Moerbeke

References

FODEH, J. S., BRANDT, C., LUONG, B. T., HADDAD, A., SCHULTZ, M., MURPHY, T., KRAUTHAMMER, M. (2013). Complementary Ensemble Clustering of Biomedical Data. J Biomed Inform. 46(3) pp.436-443.

See Also

ADEC,ADECa,ADECc

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)
MCF7_ADECb=ADECb(L,distmeasure="tanimoto",normalize=FALSE,method=NULL,
nrclusters=seq(5,25),clust="agnes",linkage="ward",alpha=0.625)
```

ADECc *Aggregated Data Ensemble Clustering - version c*

Description

Function ADECc performs aggregated data ensemble clustering in which in every iteration the number of random samples taken is randomly set between m/2 and m-1 with m the total number of features. The number of features to sample can also be prespecified by the user. Further, each resulting dendrogram is cut numerous times into a different specific number of clusters.

Usage

```
ADECc(List, distmeasure = "tanimoto",normalize=FALSE,method=NULL,t = 10,
r = NULL,nrclusters = seq(5, 25, 1), 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.
distmeasure	The distance measure to be used on the fused data matrix (character). Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. 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.
t	The number of iterations.
r	Optional. The number of features to take for the random sample.
nrclusters	A sequence of numbers of clusters to cut the dendrogram in.
clust	Choice of clustering function (character). Defaults to "agnes".
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"

Details

ADECc starts with the merging of the data matrices into one larger data matrix. Then, ensemble clustering is performed on the fused data. This comes down to repeatedly applying hierarchical clustering. A random sample of features is taken in each application. Further, variation is inserted by not splitting the dendrogram a single time into one specific number of clusters but multiple times and for a range of numbers of clusters. More information can be found in Fodeh et al. (2013).

Value

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

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

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

Note

For now, only hierarchical clustering with the `agnes` function is implemented.

Author(s)

Marijke Van Moerbeke

References

FODEH, J. S., BRANDT, C., LUONG, B. T., HADDAD, A., SCHULTZ, M., MURPHY, T., KRAUTHAMMER, M. (2013). Complementary Ensemble Clustering of Biomedical Data. J Biomed Inform. 46(3) pp.436-443.

See Also

`ADEC`,`ADECa`,`ADECb`

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)
MCF7_ADECc=ADECc(L,distmeasure="tanimoto",normalize=FALSE,method=NULL,t=10,r=NULL,
nrclusters=seq(5,25,1),clust="agnes",linkage="ward",alpha=0.625)
```

`BinFeaturesPlot` *Plot of a selection of features*

Description

The function `BinFeaturesPlot` plots the binary data matrix for a selection of features. It is possible to separate between compounds of interest and the other compounds. This is a visualization to see which characteristics are (not) expressed in a specific cluster.

Usage

```
BinFeaturesPlot(LeadCpds, OrderLab, Features, Data, ColorLab, nrclusters = NULL,
cols = NULL, name = c("FP"),colors1 = c("gray90", "blue"), colors2 = c("gray90",
"green"),margins=c(5.5,3.5,0.5,5.5),plottype="new",location=NULL)
```

Arguments

`LeadCpds`	A character vector containing the compounds one wants to separate from the others.
`Features`	A character vector containing the selection of features to be plotted.
`Data`	The data matrix the features are derived from.
`OrderLab`	Optional. If the compounds are to set in a specific order of a specific clustering.
`ColorLab`	Optional. The clustering result that determines the clusters of the labels of the objects in the plot.
`nrclusters`	Optional. The number of clusters to consider if ColorLab is specified.
`cols`	The colors for the clusters of the labels of the objects as determined by ColorLab.
`name`	The overall name to give the features.
`colors1`	Colors to indicate the present and abscence of features of the compounds not in LeadComps.
`colors2`	Colors to indicate the present and abscence of features of the LeadComps.
`margins`	Optional. Margins to be used for the plot.
`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.
`location`	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

Value

A plot indicating the values of the features of the LeadCpds in green and those of the others in blue.

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="ward",gap=FALSE,maxK=55)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",gap=FALSE,maxK=55)
## Not run:
MCF7_Interactive=ChooseFeatures(Interactive=TRUE,LeadCpds=NULL,ClusterResult=MCF7_T,
ClusterColors=MCF7_F,BinData=list(fingerprintMat),Datanames=c("FP"),geneMat,
topChar = 20, topG = 20,nrclusters=7,N=1)

Lead=MCF7_Interactive$"Choice 1"$Compounds$LeadCpds
Feat=MCF7_Interactive$"Choice 1"$Characteristic$FP

BinFeaturesPlot(LeadCpds=Lead,Features=Feat,Data=fingerprintMat,OrderLab=MCF7_F,ColorLab=MCF7_F,
nrclusters=7,cols=Colors1,name=c("FP"),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="tanimoto",
normalize=FALSE,method=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	Type indicates the kind of data provided as input. Should be one of "data", "cluster" or "distance". The type "cluster" is used if the data is the output of one of the integrated data cluster functions of the package.
distmeasure	Choice of metric for the dissimilarity matrix (character) and should only be specified if type is "data". Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. 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.
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.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

Value

One/multiple box plots.

Author(s)

Marijke Van Moerbeke

Examples

```
data(fingerprintMat)
data(targetMat)

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

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


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

CEC *Complementary Ensemble Clustering*

Description

Function CEC performs which of the function CECa, CECb or CECc is specified by the user.

Usage

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

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
distmeasure	A character vector with the distance measure for each data matrix. Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. 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.
t	The number of iterations.
r	Optional. The number of features to take for the random sample.
nrclusters	The number of clusters to cut the dendrogram in. If a sequence is specified either ADECb or ADECc is performed. A fixed number of clusters defaults to ADECa
weight	Optional. A list of different weight combinations for the data sets in List. If NULL, the weights are determined to be rqual 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. An example is provided in the details.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	A vector with the choice of inter group dissimilarity (character) for each data set.
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.

`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.

`ResampleFeatures` Logical. Whether the features should be resamples. If TRUE, either CECa or CECc is performed.

Details

See the functions for CECa, CECb and CECc for more 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 with the following four elements.

`Incidence` The summed incidence matrices for each data matrix

`IncidenceComb` The co-association matrix after a weighted sum of the elements of Incidence for each weight

`Results` The hierarchical clustering result for each element in IncidenceComb

`Clust` The result for the weight specified in Clustweight

The value has class 'CEC'

Note

For now, only hierarchical clustering with the `agnes` function is implemented.

Author(s)

Marijke Van Moerbeke

References

FODEH, J. S., BRANDT, C., LUONG, B. T., HADDAD, A., SCHULTZ, M., MURPHY, T., KRAUTHAMMER, M. (2013). Complementary Ensemble Clustering of Biomedical Data. J Biomed Inform. 46(3) pp.436-443.

See Also

CECa,CECb,CECc

Examples

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

MCF7_CECa=CEC(List=L,distmeasure=c("tanimoto","tanimoto"),
normalize=FALSE,method=NULL,t=25,r=NULL,nrclusters=c(7,7),
clust="agnes",linkage=c("flexible","flexible"),StopRange=FALSE,ResampleFeatures=TRUE)
```

CECa *Complementary Ensemble Clustering - version a*

Description

Function CECa performs complementary ensemble clustering in which in every iteration the number of random samples taken is randomly set between m/2 and m-1 with m the total number of features. The number of features to sample can also be specified by the user.

Usage

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

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
distmeasure	A character vector with the distance measure for each data matrix. Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. 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.
t	The number of iterations.
r	Optional. The number of features to take for the random sample.
nrclusters	A vector of the number of clusters to cut the dendrogram of each clustering result in.
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. An example is provided in the details.

clust	Choice of clustering function (character). Defaults to "agnes".
linkage	A vector with the choice of inter group dissimilarity (character) for each data set.
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.
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.

Details

Ensemble clustering is performed on each data matrix. This comes down to repeatedly applying hierarchical clustering. A random sample of features is taken in each application. Afterwards the incidence matrices are combined in a weighted sum and hierarchical clustering is performed once more. More information can be found in Fodeh et al. (2013).

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 the following four elements.

Incidence	The summed incidence matrices for each data matrix
IncidenceComb	The co-association matrix after a weighted sum of the elements of Incidence for each weight
Results	The hierarchical clustering result for each element in IncidenceComb
Clust	The result for the weight specified in Clustweight

The value has class 'CEC'

Note

For now, only hierarchical clustering with the agnes function is implemented.

Author(s)

Marijke Van Moerbeke

References

FODEH, J. S., BRANDT, C., LUONG, B. T., HADDAD, A., SCHULTZ, M., MURPHY, T., KRAUTHAMMER, M. (2013). Complementary Ensemble Clustering of Biomedical Data. J Biomed Inform. 46(3) pp.436-443.

See Also

CEC,CECb,CECc

Examples

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

MCF7_CECa=CECa(List=L,distmeasure=c("tanimoto","tanimoto"),
normalize=FALSE,method=NULL,t=25,r=NULL,nrclusters=c(7,7),
clust="agnes",linkage=c("flexible","flexible"),alpha=0.625,StopRange=FALSE)
```

CECb *Complementary Ensemble Clustering - version b*

Description

Function CECb performs complementary ensemble clustering in which in every iteration the total number of features are used in the clustering procedure. However, the function is capable of cutting the resulting dendrogram several times, each time into a different number of cluster.

Usage

```
CECb(List, distmeasure = c("tanimoto", "tanimoto"),normalize=FALSE,method=NULL,
nrclusters = seq(5, 25, 1), weight = NULL, clust = "agnes",
linkage=c("flexible","flexible"), alpha=0.625,WeightClust = 0.5,StopRange=FALSE)
```

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
distmeasure	A character vector with the distance measure for each data matrix. Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. 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.
nrclusters	A sequence of numbers of clusters to cut the dendrogram in.

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. An example is provided in the details.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	A vector with the choice of inter group dissimilarity (character) for each data set.
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.
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.

Details

Ensemble clustering is performed on each data matrix. This comes down to repeatedly applying hierarchical clustering. All features will be used in every iteration. Variation is inserted by not splitting the dendrogram a single time into one specific number of clusters but multiple times and for a range of numbers of clusters. Afterwards the two incidence matrices are combined in a weighted sum and hierarchical clustering is performed once more. More information can be found in Fodeh et al. (2013).

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 the following four elements.

Incidence	The summed incidence matrices for each data matrix
IncidenceComb	The co-association matrix after a weighted sum of the elements of Incidence for each weight
Results	The hierarchical clustering result for each element in IncidenceComb
Clust	The result for the weight specified in Clustweight

The value has class 'CEC'

Note

For now, only hierarchical clustering with the agnes function is implemented.

Author(s)

Marijke Van Moerbeke

References

FODEH, J. S., BRANDT, C., LUONG, B. T., HADDAD, A., SCHULTZ, M., MURPHY, T., KRAUTHAMMER, M. (2013). Complementary Ensemble Clustering of Biomedical Data. J Biomed Inform. 46(3) pp.436-443.

See Also

CEC,CECa,CECc

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)
MCF7_CECb=CECb(L,distmeasure=c("tanimoto","tanimoto"),normalize=FALSE,method=NULL,
nrclusters=seq(5,25,1),clust="agnes",linkage=c("flexible","flexible"),alpha=0.625,StopRange=FALSE)
```

CECc *Complementary Ensemble Clustering - version c*

Description

Function CECc performs complementary ensemble clustering in which in every iteration the number of random samples taken is randomly set between m/2 and m-1 with m the total number of features. The number of features to sample can also be specified by the user. Further, each resulting dendrogram can be cut numerous times into a different specific number of clusters.

Usage

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

Arguments

List	A list of data matrices. It is assumed the rows are corresponding with the objects.
distmeasure	A character vector with the distance measure for each data matrix. Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. 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.

t	The number of iterations.
r	Optional. The number of features to take for the random sample.
nrclusters	A sequence of numbers of clusters to cut the dendrogram in.
weight	Optional. A list of different weight combinations for the data sets in List. If NULL, the weights are determined to be rqual 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. An example is provided in the details.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	A vector with the choice of inter group dissimilarity (character) for each data set.
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.
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.

Details

Ensemble clustering is performed on each data matrix. This comes down to repeatedly applying hierarchical clustering. A random sample of features is taken in each application. Further, variation is inserted by not splitting the dendrogram a single time into one specific number of clusters but multiple times and for a range of numbers of clusters.Afterwards the two incidence matrices are combined in a weighted sum and hierarchical clustering is performed once more. More information can be found in Fodeh et al. (2013).

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 the following four elements.

Incidence	The summed incidence matrices for each data matrix
IncidenceComb	The co-association matrix after a weighted sum of the elements of Incidence for each weight
Results	The hierarchical clustering result for each element in IncidenceComb
Clust	The result for the weight specified in Clustweight

The value has class 'CEC'

Note

For now, only hierarchical clustering with the agnes function link is implemented.

Author(s)

Marijke Van Moerbeke

References

FODEH, J. S., BRANDT, C., LUONG, B. T., HADDAD, A., SCHULTZ, M., MURPHY, T., KRAUTHAMMER, M. (2013). Complementary Ensemble Clustering of Biomedical Data. J Biomed Inform. 46(3) pp.436-443.

See Also

CEC,CECa,CECc

Examples

```
data(fingerprintMat)
data(targetMat)
L=list(fingerprintMat,targetMat)
MCF7_CECc=CECc(L,distmeasure=c("tanimoto","tanimoto"),normalize=FALSE,method=NULL,
t=10,r=NULL,nrclusters=seq(5,25,1),clust="agnes",linkage=c("flexible","flexible")
,alpha=0.625,StopRange=FALSE)
```

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,ContData = NULL,
Datanames=NULL,nrclusters=NULL,sign=0.05,topC=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 compounds, this selection can be provided here. Selection can be of the type "character" (names of the compounds) or "numeric" (the number of specific cluster).
BinData	A list of the binary feature data matrices. These will be evaluated with the fisher's extact test.
ContData	A list of continuous data sets of the compounds. These will be evaluated with the t-test.
Datanames	A vector with the names of the binary data matrices.
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 compounds which is known by name. Otherwise, it is required.
sign	The significance level to be handled.
topC	Overrules sign. The number of features to display for each cluster. If not specified, only the significant genes are shown.
fusionsLog	To be handed to ReorderToReference.
WeightClust	To be handed to ReorderToReference.
names	Optional. Names of the methods.

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:

Compounds	A list with the elements LeadCpds (the compounds of interest) and OrderedCpds (all compounds 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)

Author(s)

Marijke Van Moerbeke

Examples

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

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",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("F","T"),nrclusters=7,topC=NULL,sign=0.05,fusionsLog=TRUE,WeightClust=TRUE,
names=c("F","T"))

## 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 compounds of interest without producing the plot.

Usage

```
ChooseCluster(Interactive = TRUE, LeadCpds=NULL, ClusterResult, ColorLab
= NULL,BinData=NULL,ContData=NULL,Datanames = c("FP"), GeneExpr, topChar = 20, topG = 20,
sign = 0.05, nrclusters = NULL, cols = NULL, N = 1)
```

Arguments

`Interactive`	Logical. Produce plot or not. Defaults to TRUE.
`LeadCpds`	A list of the compounds 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.
`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.

ContData	A list of continuous data sets of the compounds. These will be evaluated with the t-test.
Datanames	A character vector of the labels to give to the matrices in BinData.
GeneExpr	A gene expression matrix, may also be an ExpressionSet. The rows should correspond with the genes.
topChar	The number of top characteristics to return. If NULL, only the significant characteristics are saved.
topG	The number of top genes to return. If NULL, only the significant genes are saved.
sign	The significance level.
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.
cols	The colors to use in the dendrogram.
N	The number of clusters one wants to identify by a mouse click.

Details

The DE genes are determined by testing for significance of the specified cluster versus all other compounds 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:

Compounds	A list with the elements LeadCpds (the compounds of interest) and OrderedCpds (all compounds 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)

Author(s)

Marijke Van Moerbeke

Examples

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

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

```
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",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"),geneMat,
topChar = 20, topG = 20,nrclusters=7,N=1)

## End(Not run)
```

`Cluster` *Perform clustering on a single data source*

Description

The function `Cluster` was written to perform 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 ="ward",alpha=0.625
,gap = TRUE,maxK = 50,StopRange=FALSE)
```

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".
`normalize`	Logical. Indicates whether to normalize the distance matrices or not. 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.
`clust`	Choice of clustering function (character). Defaults to "agnes".
`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"
`gap`	Logical. Indicator if gap statistics should be computed. Setting to $FALSE$ will greatly reduce the computation time.

maxK	The maximum number of clusters to be considered during the gap.
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.

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.

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.

Author(s)

Marijke Van Moerbeke

Examples

```
data(fingerprintMat)
data(targetMat)

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

`ClusterPlot` *Plot a dendrogram with leaves colored by a result of choice*

Description

The above described function `ClusterCols` is used in the function Clusterplot which actually plots the dendrogram made by ClusterCols. Further, given the outputs of any other functions, it is capable of selecting the elements needed for ClusterCols.

Usage

```
ClusterPlot(Data1, Data2=NULL, nrclusters = NULL, cols = NULL,plottype="new",
location=NULL,ColorComps = 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.
`nrclusters`	Optional. The number of clusters to cut the dendrogram in. If not specified the dendrogram will be drawn without colors to discern the different clusters.
`cols`	The colors for the clusters if nrclusters is specified.
`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.
`location`	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.
`ColorComps`	If only a specific set of compounds need to be highlighted, this can be specified here. The compounds should be given in a character vector. If specified, all other compound labels will be colored black.
`...`	Other options which can be given to the plot function.

Details

This function relies on the internal `ClusterCols` 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.

Author(s)

Marijke Van Moerbeke

Examples

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

MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",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))
```

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 compounds, colors can be used. The function `ColorPalette` is able to pick out as many colors 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.

Author(s)

Marijke Van Moerbeke

Examples

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

Colors1 *First example for colors*

Description

A data set HEX code for the colors used in the examples.

Usage

```
data("Colors1")
```

Format

The format is: chr [1:8] "#8EE5EE" "#D2691E" "#EE2C2C" "#EEAD0E" "#006400" ...

Colors2 *Second example for colors*

Description

A data set HEX code for the colors used in the examples.

Usage

```
data("Colors2")
```

Format

The format is: chr [1:8] "#D2691E" "#EE2C2C" "#EEAD0E" "#006400" "#0000EE" ...

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	The hex codes of the colors to be used.

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.

Author(s)

Marijke Van Moerbeke

See Also

ReorderToReference

Examples

```
data(fingerprintMat)
data(targetMat)
data(Colors2)

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

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

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

Names=ColorsNames(MatrixColors,cols=Colors2)
```

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

Description

A visual comparison of all methods is handy to see which compounds 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.
cols	The hex codes of the colors to be used.
fusionsLogM	The fusionsLog parameter for the elements in ListM. To be handed to ReorderToReference.
fusionsLogS	The fusionslog parameter for the elements in ListS. To be handed to ReorderToReference.
WeightClustM	The WeightClust parameter for the elements in ListM. To be handed to ReorderToReference.
WeightClustS	The WeightClust parameter for the elements in ListS. To be handed to ReorderToReference.
namesS	Optional. Names of the single source clusterings to be used as labels for the columns.
namesM	Optional. Names of the multiple source clusterings to be used as labels for the columns.
marginsM	Optional. Margins to be used for the plot for the elements is ListM after the identification.
marginsS	Optional. Margins to be used for the plot for the elements is ListS after the identification.
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.
N	The number of methods/clusters you want to identify.
...	Other options which can be given to the color2D.matplot function.

Details

This function relies on ComparePlot to plot the results.

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.

Author(s)

Marijke Van Moerbeke

See Also

ComparePlot

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(Colors2)

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

L=list(fingerprintMat,targetMat)

MCF7_W=WeightedClust(L,type="data",distmeasure=c("tanimoto","tanimoto"),normalize=FALSE,
method=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=Colors2,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 compounds will always cluster together independent of the applied methods. To this aid the function ComparePlot has been written.

Usage

```
ComparePlot(List, nrclusters = NULL, cols = NULL, fusionsLog = FALSE,
WeightClust = FALSE, names = NULL,margins = c(8.1, 3.1, 3.1, 4.1),
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.
cols	The hex codes of the colors to be used.
fusionsLog	To be handed to ReorderToReference.
WeightClust	To be handed to ReorderToReference.
names	Optional. Names of the methods to be used as labels for the columns.
margins	Optional. Margins to be used for the plot.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.
...	Other options which can be given to the color2D.matplot function.

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.

Author(s)

Marijke Van Moerbeke

See Also

ReorderToReference,ColorsNames

Examples

```
data(fingerprintMat)
data(targetMat)
data(Colors2)

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

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

ComparePlot(L,nrclusters=7,cols=Colors2,fusionsLog=TRUE,WeightClust=TRUE,names=names,
margins=c(9.1,4.1,4.1,4.1),plottype="new",location=NULL)
```

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=FALSE,method=NULL,nrclusters=NULL,names=NULL,
nboot=1000,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	Type indicates whether the provided matrices in "List" are either data matrices, distance matrices.
distmeasure	A character vector with the distance measure for each data matrix. Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.
nrclusters	The number of clusters to cut the dendrogram in. This is necessary for the computation of the Jaccard coefficient.
names	The labels to give to the elements in List.
nboot	Number of bootstraps to be run.
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.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

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 indicates 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 modeled 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

Author(s)

Marijke Van Moerbeke

Examples

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

List=list(fingerprintMat,targetMat)

Comparison=CompareSilCluster(List=List,type="data",
distmeasure=c("tanimoto","tanimoto"),normalize=FALSE,method=NULL,
nrclusters=7,names=NULL,nboot=1000,StopRange=FALSE,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 compounds 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.
`cols`	The hex codes of the colors to be used.
`fusionsLogS`	The fusionslog parameter for the elements in ListS. To be handed to `ReorderToReference`.

fusionsLogM	The fusionsLog parameter for the elements in ListM. To be handed to ReorderToReference.
WeightClustS	The WeightClust parameter for the elements in ListS. To be handed to ReorderToReference.
WeightClustM	The WeightClust parameter for the elements in ListM. To be handed to ReorderToReference.
namesS	Optional. Names of the single source clusterings to be used as labels for the columns.
namesM	Optional. Names of the multiple source clusterings to be used as labels for the columns.
margins	Optional. Margins to be used for the plot.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.
...	Other options which can be given to the color2D.matplot function.

Details

This function relies on ComparePlot to plot both the results of the single source clusterings as the multiple source clusterings.

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.

Author(s)

Marijke Van Moerbeke

See Also

ComparePlot

Examples

```
data(fingerprintMat)
data(targetMat)
data(Colors2)

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

L=list(fingerprintMat,targetMat)
```

```
MCF7_W=WeightedClust(L,type="data", distmeasure=c("tanimoto","tanimoto"),normalize=FALSE,
method=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=Colors2,fusionsLogS=FALSE,
fusionsLogM=FALSE,WeightClustS=FALSE,WeightClustM=FALSE,namesS,
namesM,reverse=FALSE,plottype="new",location=NULL)
```

`ContFeaturesPlot` *Plot of continuous features*

Description

The function `BioassayPlot` plots the bioassay values for the compounds. It is possible to separate between compounds of interest and the other compounds. This is a visualization to see how bio-assays differ from cluster to cluster.

Usage

```
ContFeaturesPlot(LeadCpds,Data,nrclusters,OrderLab,ColorLab=NULL,cols=NULL,
ylab="bio-assays",AddLegend=TRUE,margins=c(5.5,3.5,0.5,8.7),plottype="new",location=NULL)
```

Arguments

`LeadCpds`	A character vector containing the compounds one wants to separate from the others.
`Data`	The bio-assay data matrix.
`nrclusters`	Optional. The number of clusters to consider if ColorLab is specified.
`OrderLab`	Optional. If the compounds are to set in a specific order of a specific method.
`ColorLab`	The clustering result that determines the color of the labels of the objects in the plot. If NULL, the labels are black.
`cols`	The colors for the labels of the objects.
`ylab`	The lable of the y-axis.
`AddLegend`	Logical. Indicates whether a legend should be added to the plot.
`margins`	Optional. Margins to be used for the plot.
`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.
`location`	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

Value

A plot in which the values of the bio-assays for the LeadCpds are separeted from the others.

Author(s)

Marijke Van Moerbeke

Examples

```
data(Colors2)
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=Colors2,ylab="bio-assays",AddLegend=TRUE,margins=c(5.5,3.5,0.5,8.7),
plottype="new",location=NULL)
```

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"),weight=seq(0,1,by=0.01),
distmeasure=c("tanimoto","tanimoto"),normalize=FALSE,method=NULL,nrclusters=NULL,
names=NULL,nboot=1000,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	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".
weight	Optional. A list of different weight combinations for the data sets in List. If NULL, the weight is sequence from 0 to 1 with steps of 0.1 and a result is produced for each weight. It is further possible to fix weights for some data matrices and to let it vary randomly for the remaining data sets. 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.
distmeasure	A character vector with the distance measure for each data matrix. Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.
names	The labels to give to the elements in List.
nboot	Number of bootstraps to be run.
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.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

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

Author(s)

Marijke Van Moerbeke

Examples

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

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

L=list(MCF7_F,MCF7_T)

MC7_Weight=DetermineWeight_SilClust(List=L,type="clusters",weight=seq(0,1,by=0.01),nrclusters=7,
distmeasure=c("tanimoto","tanimoto"),normalize=FALSE,method=NULL,names=c("FP","TP"),nboot=100,
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"),
weight=seq(0, 1, by = 0.01),nrclusters = NULL, distmeasure = c("tanimoto",
"tanimoto"),normalize=FALSE,method=NULL,clust = "agnes",linkage=c("ward","ward")
,alpha=0.625,gap = FALSE, maxK = 50, names = c("B", "FP"),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	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".
weight	Optional. A list of different weight combinations for the data sets in List. If NULL, the weights are determined to be rqual 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. 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.
distmeasure	A character vector with the distance measure for each data matrix. Should be one of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	A vector with the choice of inter group dissimilarity (character) for each data set.

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".
maxK	The maximal number of clusters to consider in calculating the gap statistic. Only if type="data".
names	The labels to give to the elements in List.
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.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

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

Author(s)

Marijke Van Moerbeke

References

PERUALILA-TAN, N., SHKEDY, Z., TALLOEN, W., GOEHLMANN, H. W. H., QSTAR Consortium, VAN MOERBEKE, M., KASIM, A., (in press). Weighted-Similarity Based Clustering of Chemical Structure and Bioactivity Data in Early Drug Discovery. Journal of Bioinformatics and Computational Biology.

See Also

WeightedSimClust

Examples

```
data(fingerprintMat)
data(targetMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",alpha=0.625,gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",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=7,
distmeasure=c("tanimoto","tanimoto"),normalize=FALSE,method=NULL,clust="agnes",
linkage=c("flexible","flexible"),alpha=0.625,gap=FALSE,maxK=50,names=c("FP","TP"),StopRange=FALSE,
plottype="new",location=NULL)
```

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 compounds, this selection can be provided here. Selection can be of the type "character" (names of the compounds) or "numeric" (the number of specific cluster).
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 compounds which is known by name. Otherwise, it is required.
method	The method to applied to look for DE genes. For now, only the limma method is available
sign	The significance level to be handled.
topG	Overrules sign. The number of top genes to be shown.
fusionsLog	To be handed to ReorderToReference.
WeightClust	To be handed to ReorderToReference.
names	Optional. Names of the methods.

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:

Compounds	A list with the elements LeadCpds (the compounds of interest) and OrderedCpds (all compounds 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)

Author(s)

Marijke Van Moerbeke

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

L=list(MCF7_T ,MCF7_F)

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

Distance *Distance function*

Description

The Distance function was written 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".
normalize	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.

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 compounds 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.

Author(s)

Marijke Van Moerbeke

References

LI, Y., TU, K., ZHENG, S., WANG, J., LI, Y., LI, X. (2011). Association of Feature Gene Expression with Structural Fingerprints of Chemical Compounds. Journal of Bioinformatics and Computational biology. 9(4). pp. 503-519. MAECHLER, M., ROUSSEEUW, P., STRUYF, A., HUBERT, M. (2014). cluster: Cluster Analysis Basics and Extensions. R package version 1.15.3. TALLOEN, W., VERBEKE, T. (2011). a4: Automated Affymetrix Array Analysis Umbrella Package. R package version 1.14.0

Examples

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

`FeaturesOfCluster` *Lists all features present in a selected cluster of compounds*

Description

The function `FeaturesOfCluster` lists the number of features compounds of the cluster have in common. A threshold can be set selecting among how many compounds 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 compounds one wants to investigate in terms of features.
`Data`	The data matrix the features are derived from.
`Threshold`	The number of compounds 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 compounds specified in leadCpds.
`Plot`	Logical. Indicates whether or not a BinFeaturesPlot should be set up for the selectcion of compounds and discovered features.
`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.
`location`	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

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 compounds. By including all other compounds as well, one can see whether features are common in the compounds or rather specific for the cluster.

Further, it returns a list with 2 items. The first indicates the number of shared features among the compounds. This provides an overview of which compounds 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.

Author(s)

Marijke Van Moerbeke

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 compounds in the output of* ReorderToReference

---

Description

FindCluster selects the compounds 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.
select	The row (the method) and the number of the cluster to select.
fusionsLog	To be handed to ReorderToReference.
WeightClust	To be handed to ReorderToReference.
names	Optional. Names of the methods.

Value

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

Author(s)

Marijke Van Moerbeke

Examples

```
data(fingerprintMat)
data(targetMat)

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

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

Comps=FindCluster(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, Object, Element = list())
```

Arguments

`What`	A character string indicating which object to look for.
`Object`	The data structure to look into. Only the classes data frame and list are supported.
`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.

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="ward",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",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('TopDE',MCF7_DiffGenes_FandT10)
```

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

Description

Due to the shifting of compounds 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.

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="ward",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",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 *The fingerprint matrix for the MCF7 data*

Description

A binary data set that contains 250 fingerprints for the CMAP MCF7 data.

Usage

```
data("fingerprintMat")
```

Format

The format is: logi [1:56, 1:250] FALSE FALSE FALSE FALSE FALSE FALSE ... - attr(*, "dimnames")=List of 2 ..$: chr [1:56] "metformin" "phenformin" "phenyl biguanide" "estradiol" ... ..$: chr [1:250] "-2147375257" "-2147119955" "-2146474760" "-2145840573" ...

GeneInfo *The gene info data frame*

Description

The data set contains the entrezIdentifiers, symbols and names of the used genes in the CMAP MCF7 data.

Usage

```
data("GeneInfo")
```

Format

The format is: 'data.frame': 2434 obs. of 3 variables: $ ENTREZID: Factor w/ 2434 levels "10001","100129361",..: 1 2 3 4 5 6 7 8 9 10 ... ..- attr(*, "names")= chr "1" "2" "3" "4" ... $ SYMBOL : Factor w/ 2434 levels "AARS","ABCA1",..: 1213 1132 1486 2178 1914 1133 2175 891 2003 1134 ... ..- attr(*, "names")= chr "1" "2" "3" "4" ... $ GENENAME: Factor w/ 2434 levels "1-acylglycerol-3-phosphate O-acyltransferase 5 (lysophosphatidic acid acyltransferase, epsilon)",..: 1199 925 1586 2153 1895 2374 2067 908 1983 926 ... ..- attr(*, "names")= chr "1" "2" "3" "4" ...

---

geneMat *The gene expression matrix*

Description

The gene expression data for 2434 genes for the compounds in the CMAP MCF7 data.

Usage

```
data("geneMat")
```

Format

The format is: num [1:2434, 1:56] -0.0772 -0.0698 -0.055 -0.0498 -0.0597 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:2434] "MED6" "LOC100129361" "PDCD6IP" "TOMM6" ... ..$: chr [1:56] "metformin" "phenformin" "phenyl biguanide" "estradiol" ...

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

Description

The function Geneset.intersect puts per method the results of the PathwaysIter function together 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 compounds. If this was the case, Selection should be put to TRUE. Otherwise, it should be put to FALSE.
sign	The significance level to be handled for cutting of the pathways.
names	Optional. Names of the methods.
seperatetables	Logical. If TRUE, a separate element is created per cluster. containing the pathways for each iteration.
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.

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.

Author(s)

Marijke Van Moerbeke

See Also

PathwaysIter

Examples

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

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

L=list(MCF7_F,MCF7_T)

MCF7_Paths_FandT=PathwaysIter(list(MCF7_F,MCF7_T),GeneExpr=geneMat,nrclusters=7,
method=c("limma", "MLP"),ENTREZID=GeneInfo[,1],geneSetSource = "GOBP",top=NULL,
topG=NULL,GENESET=ListGO,sign=0.05,niter=2,fusionsLog=TRUE,WeightClust=TRUE,
names=c("FP","TP"))

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

```
str(MCF7_Paths_intersection)

## End(Not run)
```

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.

Usage

```
data(GS)
```

Format

The format is: List of 8734 $ GO:0000002: chr [1:20] "291" "1763" "1890" "3980" ... $ GO:0000003: chr [1:925] "18" "49" "51" "90" ... $ GO:0000012: chr [1:8] "3981" "7141" "7515" "23411" ... $ GO:0000018: chr [1:51] "604" "641" "940" "958" ... $ GO:0000019: chr [1:4] "641" "4292" "4361" "10111" $ GO:0000022: chr [1:2] "9055" "9493" ... $ GO:0000724: chr [1:70] "472" "641" "672" "675" ... [list output truncated] - attr(*, "species")= chr "Human" - attr(*, "geneSetSource")= chr "GOBP" - attr(*, "descriptions")= Named chr [1:13226] "mitochondrial genome maintenance" "reproduction" "single strand break repair" "regulation of DNA recombination" ... ..- attr(*, "names")= chr [1:13226] "GO:0000002" "GO:0000003" "GO:0000012" "GO:0000018" ... - attr(*, "class")= chr [1:2] "geneSetMLP" "list"

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.

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.
nrclusters	The number of clusters to cut the dendrogram in.
cols	The colors to be used for the clusters.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

Details

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.

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 compounds 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.

Author(s)

Marijke Van Moerbeke

Examples

```
data(fingerprintMat)
data(targetMat)
data(Colors2)

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

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

HeatmapPlot(MCF7_T,MCF7_F,names=rownames(fingerprintMat),nrclusters=7,cols=Colors2,plottype="new",
location=NULL)
```

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

Description

The function HeatmapSelection plots the similarity values between compounds. 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 compounds of interest. See more in Details.

Usage

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

Arguments

Data	The data of which a heatmap should be drawn.
type	The type of data. Data can either be the data itself ("data"), the outcome of a clustering method ("clust'), a distance matrix ("dist") or a similarity matrix ("sim").
distmeasure	If type is "data", a distance measure for the clustering should be specified.
normalize	Logical. If type is "data", it can be specified whether the data should be normalized.
method	If type is "data" and normalize is TRUE, a method for normalization should be specified. See Normalization.
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 compounds.
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.
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 compounds in their order of clustering on the plot.

width	The width of the plot to be made. This can be adjusted since the default size might not show a clear picture.
height	The height of the plot to be made. This can be adjusted since the default size might not show a clear picture.

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 compounds. 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 compounds 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.

Value

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

Author(s)

Marijke Van Moerbeke

Examples

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

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",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)
```

LabelPlot *Coloring specific leaves of a dendrogram*

Description

Just as the function ClusterCols, LabelCols as its own plotting function LabelPlot which plots the dendrogram.

Usage

```
LabelPlot(Data, Sel1, Sel2 = NULL, col1, col2 = NULL, ...)
```

Arguments

Data	The result of a method which contains the dendrogram to be colored.
Sel1	The selection of objects to be colored.
Sel2	An optional second selection to be colored.
col1	The color for the first selection.
col2	The color for the optional second selection.
...	Other options which can be given to the plot function.

Value

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

Author(s)

Marijke Van Moerbeke

Examples

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

ClustF_6=cutree(MCF7_F$Clust,6)

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

LabelPlot(MCF7_F,Sel1=SelF,Sel2=NULL,col1='darkorchid')
```

Normalization *A normalization function*

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. This is performed by the Normalization function.

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 distance matrix.

Author(s)

Marijke Van Moerbeke

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 compounds, this selection can be provided here. Selection can be of the type "character" (names of the compounds) or "numeric" (the number of specific cluster).
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.
method	The method to applied to look for DE genes. For now, only the limma method is available.
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.
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.
topG	Overrules sign. The number of top genes to be returned in the result. If not specified, only the significant genes are shown.
GENESET	Optional. Can provide own candidate gene sets.
sign	The significance level to be handled.
niter	The number of times to perform pathway analysis.
fusionsLog	To be handed to ReorderToReference.
WeightClust	To be handed to ReorderToReference.
names	Optional. Names of the methods.
seperatetables	Logical. If TRUE, a separate element is created per cluster. containing the pathways for each iteration.
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.

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.

Author(s)

Marijke Van Moerbeke

Examples

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

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

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

MCF7_PathsFandT=PathwaysAnalysis(L, GeneExpr = geneMat, nrclusters = 7, method = c("limma",
"MLP"), GeneInfo = GeneInfo, geneSetSource = "GOBP", topP = NULL,
topG = NULL, GENESET = GS, 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 the 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 the`DiffGenes` 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 compounds, this selection can be provided here. Selection can be of the type "character" (names of the compounds) or "numeric" (the number of specific cluster).
`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 compounds which is known by name. Otherwise, it is required.
method	The method to applied to look for DE genes. For now, only the limma method is available.
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.
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.
topG	Overrules sign. The number of top genes to be returned in the result. If not specified, only the significant genes are shown.
GENESET	Optional. Can provide own candidate gene sets.
sign	The significance level to be handled.
fusionsLog	To be handed to ReorderToReference.
WeightClust	To be handed to ReorderToReference.
names	Optional. Names of the methods.

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:

Compounds	A list with the elements LeadCpds (the compounds of interest) and OrderedCpds (all compounds 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.

Author(s)

Marijke Van Moerbeke

See Also

PathwaysIter

Examples

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

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

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

MCF7_PathsFandT=Pathways(L, GeneExpr = geneMat, nrclusters = 7, method = c("limma",
"MLP"), GeneInfo = GeneInfo, geneSetSource = "GOBP", topP = NULL,
topG = NULL, GENESET = GS, 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 compounds, this selection can be provided here. Selection can be of the type "character" (names of the compounds) or "numeric" (the number of specific cluster).
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.
method	The method to applied to look for DE genes. For now, only the limma method is available.
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.
geneSetSource	The source for the getGeneSets function ("GOBP", "GOMF","GOCC", "KEGG" or "REACTOME").
topP	Overrules sign. The number of pathways to display for each cluster. If not specified, only the significant genes are shown.
topG	Overrules sign. The number of top genes to be returned in the result. If not specified, only the significant genes are shown.
GENESET	Optional. Can provide own candidate gene sets.
sign	The significance level to be handled.
niter	The number of times to perform pathway analysis.
fusionsLog	To be handed to ReorderToReference.
WeightClust	To be handed to ReorderToReference.
names	Optional. Names of the methods.

Value

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

Compounds	A list with the elements LeadCpds (the compounds of interest) and OrderedCpds (all compounds 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.

Author(s)

Marijke Van Moerbeke

Examples

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

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

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

MCF7_PathsFandT=PathwaysIter(L, GeneExpr = geneMat, nrclusters = 7, method = c("limma",
"MLP"), GeneInfo = GeneInfo, geneSetSource = "GOBP", topP = NULL,
topG = NULL, GENESET = GS, sign = 0.05,niter=2,fusionsLog = TRUE, WeightClust = TRUE,
 names =names)

## End(Not run)
```

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

Description

The PlotPathways function takes an output of the PathwayAnalysis fucntion 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
main	Title of the plot.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

Value

The output is a GO graph.

Author(s)

Marijke Van Moerbeke

Examples

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

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

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

MCF7_PathsFandT=PathwayAnalysis(L, GeneExpr = geneMat, nrclusters = 7, method = c("limma",
"MLP"), ENTREZID = GeneInfo[, 1], geneSetSource = "GOBP", topP = NULL,
topG = NULL, GENESET = GS, sign = 0.05,niter=2,fusionsLog = TRUE, WeightClust = TRUE,
 names =names,seperatetables=FALSE,separatepvals=FALSE)

PlotPathways(MCF7_PathsFandT$FP$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 "Compounds" such that the function knows which compounds 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.
`sign`	The significance level to be handled.

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 compounds.
`Compounds`	This is a list with that contains another list per group of tested compounds. Every list contains the lead compounds and the ordered compounds.
`Genes`	This is a list with that contains contains another list per group of tested compounds. Every list contains two data frames, one with information on the top genes and one with information on all genes.

Author(s)

Marijke Van Moerbeke

Examples

```
data(fingerprintMat)
data(geneMat)

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",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$'Cluster 1'$Compounds$LeadCpds=Comps1
L2$'Cluster 2'$Compounds$LeadCpds=Comps2
L2$'Cluster 3'$Compounds$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?
OrderLab	Optional. If the compounds are to set in a specific order of a specific method.

ColorLab	The clustering result that determines the color of the labels of the objects in the plot.
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.
extra	The space between the plot and the legend.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.
...	Optional parameter to be handed to the plot function.

Value

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

Author(s)

Marijke Van Moerbeke

See Also

ProfilePlot

Examples

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

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

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

MCF7_Paths_FandT=PathwaysIter(L,GeneExpr=geneMat,nrclusters=7,method=c("limma", "MLP"),
GeneInfo=GeneInfo,geneSetSource = "GOBP",top=NULL,topG=NULL,GENESET=ListGO,sign=0.05,
```

```
niter=2,fusionsLog=TRUE,WeightClust=TRUE,names=c("FP","TP"))

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),geneMat,nrclusters=7,"limma",0.05,top=10)

MCF7_Shared10=Shared(DataLimma=MCF7_DiffGenes_FandT10,DataMLP=MCF7_Paths_intersection)

Comps=SharedComps(list(MCF7_DiffGenes_FandT10$`Method 1`$"Cluster 1",
MCF7_DiffGenes_FandT10$`Method 2`$"Cluster 1"))

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

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

ListC=list(MCF7_DiffGenes_FandT10[[1]][[1]]$Compounds$LeadCpds,
MCF7_DiffGenes_FandT10[[2]][[1]]$Compounds$LeadCpds)

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,nrcluster=7,Clusters=ListC,cols=colscl,AddLegend=TRUE,
usedgenes=Genes,margins=c(8.1,4.1,1.1,6.5),plottype="new",location=NULL,cex=0.75)

## 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.
`fusionsLog`	Logical indicator for the fusion of clusters.
`WeightClust`	Optional. To be used for the outputs of CEC, WeightedClust or WeightedSimCLust. Then only the result of the Clust element is considered.
`names`	Optional. 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 compounds 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 compounds into other clusters.

Value

A matrix of which the cells indicate to what cluster the compounds 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.

Author(s)

Marijke Van Moerbeke

Examples

```
data(fingerprintMat)
data(targetMat)

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

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

MCF7_Matrix=ReorderToReference(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.

Usage

```
SelectnrClusters(List,type=c("data","dist","pam"),distmeasure=c("tanimoto","tanimoto")
,normalize=FALSE,method=NULL,nrclusters = seq(5, 25, 1),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`	Type indicates whether the provided matrices in "List" are either data matrices, distance matrices or clustering results obtained with `pam` of the **cluster** package. Type should be one of "data", "dist" or"pam".
`distmeasure`	A character vector with the distance measure for each data matrix. Should be one of "tanimoto", "euclidean", "jaccard","hamming".
`normalize`	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.
`nrclusters`	A sequence of numbers of clusters to cut the dendrogram in.
`names`	The labels to give to the elements in List.
`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.
`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.
`location`	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

Details

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.

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

Author(s)

Marijke Van Moerbeke

Examples

```
data(fingerprintMat)
data(targetMat)

List=list(fingerprintMat,targetMat)

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

NrClusters
```

SharedComps *Intersection of clusters over multiple methods*

Description

The SharedComps function is an easy way to select the compounds 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.
fusionsLog	To be handed to ReorderToReference.
WeightClust	To be handed to ReorderToReference.
names	Names of the methods or clusters.

Value

A vector containing the shared compounds of all listed elements.

Author(s)

Marijke Van Moerbeke

Examples

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

MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",gap=FALSE,maxK=55,StopRange=FALSE)
MCF7_T = Cluster(targetMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",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 compounds 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 compounds 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.
DataMLP	Optional. The output of Geneset.intersect function.
DataFeat	Optional. The output of CharacteristicFeatures function.
names	Optional. Names of the methods or "Selection" if it only considers a selection of compounds.
Selection	Logical. Do the results concern only a selection of compounds or a specific cluster? If yes, then Selection should be put to TRUE. Otherwise all compounds and clusters are considered.

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 compounds 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.

Author(s)

Marijke Van Moerbeke

Examples

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

```
data(GeneInfo)
data(ListGO)

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

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

MCF7_Paths_FandT=PathwaysIter(L,GeneExpr=geneMat,nrclusters=7,method=c("limma", "MLP"),
ENTREZID=GeneInfo[,1],geneSetSource = "GOBP",top=NULL,topG=NULL,GENESET=ListGO,sign=0.05,
niter=2,fusionsLog=TRUE,WeightClust=TRUE,names=c("FP","TP"))

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),geneMat,nrclusters=7,"limma",0.05,top=10)

MCF7_Char=CharacteristicFeatures(list(MCF7_F,MCF7_T),Selection=NULL,BinData=list(fingerprintMat,
targetMat),Datanames=c("F","T"),nrclusters=7,top=NULL,sign=0.05,fusionsLog=TRUE,WeightClust=TRUE,
names=c("F","T"))

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

## End(Not run)
```

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

Description

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

Usage

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

Arguments

`Data`	The data of which a heatmap should be drawn.

type	The type of data. Data can either be the data itself ("data"), the outcome of a clustering method ("clust'), a distance matrix ("dist") or a similarity matrix ("sim").
distmeasure	If type is "data", a distance measure for the clustering should be specified.
normalize	Logical. If type is "data", it can be specified whether the data should be normalized.
method	If type is "data" and normalize is TRUE, a method for normalization should be specified. See Normalization.
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 compounds.
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.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

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 compounds on the right and bottom and a dendrogram of the clustering at the left and top.

Author(s)

Marijke Van Moerbeke

Examples

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


MCF7_F = Cluster(fingerprintMat,type="data",distmeasure="tanimoto",normalize=FALSE,
method=NULL,clust="agnes",linkage="ward",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.~~
fusionsLog	To be handed to MatrixFunction.
WeightClust	To be handed to MatrixFunction.
names	Optional. Names of the methods.

Value

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

Author(s)

Marijke Van Moerbeke

Examples

```
data(fingerprintMat)
data(targetMat)

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

```
method=NULL,clust="agnes",linkage="ward",gap=FALSE,maxK=55,StopRange=FALSE)

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

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

SNF *Similarity Network Fusion*

Description

The function SNF performs one of the functions SNFa, SNFb or SNFc as specified by the user.

Usage

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

Arguments

List	A list of matrices of the same type. It is assumed the rows are corresponding with the objects.
type	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 of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.
NN	The number of neighbours to be used in the procedure.
mu	The parameter epsilon. The value is recommended to be between 0.3 and 0.8.
T	The number of iterations.
clust	Choice of clustering function (character). Defaults to "agnes".
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"

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.
Version	Specifies which version of SNF to perform. Should be one of "SNFa", "SNFb" or "SNFc".

Value

The returned value is a list with two elements:

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

Note

For now, only hierarchical clustering with the agnes function is implemented.

Author(s)

Marijke Van Moerbeke

References

WANG, B., MEZLINI, M. A., DEMIR, F., FIUME, M., TU, Z., BRUDNO, M., HAIBE-KAINS, B., GOLDENBERG, A. (2014). Similarity Network Fusion for aggregating data types on a genomic scale. Nature. 11(3) pp. 333-337.

See Also

SNFa,SNFb,SNFc

Examples

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

MCF7_SNFa=SNF(list(fingerprintMat,targetMat),type="data",distmeasure=c("tanimoto",
"tanimoto"),normalize=FALSE,method=NULL,NN=10,mu=0.5,T=20,clust="agnes",linkage="ward"
,alpha=0.625,StopRange=FALSE,Version="SNFa")
```

SNFa *Similarity Network Fusion - version a*

Description

The function SNFa performs similarity network fusion as implemented by the package SNFtool. The overall method is described in the paper by Wang et al (2014).

Usage

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

Arguments

List	A list of matrices of the same type. It is assumed the rows are corresponding with the objects.
type	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 of "tanimoto", "euclidean", "jaccard","hamming".
normalize	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.
NN	The number of neighbours to be used in the procedure.
mu	The parameter epsilon. The value is recommended to be between 0.3 and 0.8.
T	The number of iterations.
clust	Choice of clustering function (character). Defaults to "agnes".
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"
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.

Value

The returned value is a list with two elements:

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

Note

For now, only hierarchical clustering with the agnes function is implemented.

Author(s)

Marijke Van Moerbeke

References

WANG, B., MEZLINI, M. A., DEMIR, F., FIUME, M., TU, Z., BRUDNO, M., HAIBE-KAINS, B., GOLDENBERG, A. (2014). Similarity Network Fusion for aggregating data types on a genomic scale. Nature. 11(3) pp. 333-337.

See Also

SNF,SNFb,SNFc

Examples

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

MCF7_SNFa=SNFa(list(fingerprintMat,targetMat),type="data",distmeasure=c("tanimoto",
"tanimoto"),normalize=FALSE,method=NULL,NN=10,mu=0.5,T=20,clust="agnes",linkage="ward"
,alpha=0.625,StopRange=FALSE)
```

SNFb *Similarity Network Fusion - version b*

Description

Function SNFb performs SNF but first determines the subsets of neighbours and then normalization is performed on the neighbours only. The function is based on the functions affinityMatrix and snf from the SNFtool package.

Usage

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

Arguments

List	A list of matrices of the same type. It is assumed the rows are corresponding with the objects.
type	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 of "tanimoto", "euclidean", "jaccard","hamming".
method	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names.
normalize	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization in Normalization.
NN	The number of neighbours to be used in the procedure.
mu	The parameter epsilon. The value is recommended to be between 0.3 and 0.8.
T	The number of iterations.
clust	Choice of clustering function (character). Defaults to "agnes".
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"
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.

Value

The returned value is a list with two elements:

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

Note

For now, only hierarchical clustering with the agnes function link is implemented.

Author(s)

Marijke Van Moerbeke

References

WANG, B., MEZLINI, M. A., DEMIR, F., FIUME, M., TU, Z., BRUDNO, M., HAIBE-KAINS, B., GOLDENBERG, A. (2014). Similarity Network Fusion for aggregating data types on a genomic scale. Nature. 11(3) pp. 333-337. WANG, B., MEZLINI, M. A., DEMIR, F., FIUME, M., TU, Z., BRUDNO, M., HAIBE-KAINS, B., GOLDENBERG, A. (2014). SNFtool: Similarity Network Fusion. R package version 2.2

See Also

SNF,SNFa,SNFc

Examples

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

MCF7_SNFb=SNFb(list(fingerprintMat,targetMat),type="data",distmeasure=c("tanimoto",
"tanimoto"),normalize=FALSE,method=NULL,NN=10,mu=0.5,T=20,clust="agnes",linkage="ward"
,alpha=0.625,StopRange=FALSE)
```

SNFc *Similarity Network Fusion - version c*

Description

Function SNFc performs SNF but first a normalization over all objects is performed before taking the k neighbours of each object as a subset in obtaining the kernel matrix. The function is based on the functions affinityMatrix and snf from the SNFtool package.

Usage

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

Arguments

List	A list of matrices of the same type.It is assumed the rows are corresponding with the objects.
type	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 of "tanimoto", "euclidean", "jaccard","hamming".

normalize	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.
NN	The number of neighbours to be used in the procedure.
mu	The parameter epsilon. The value is recommended to be between 0.3 and 0.8.
T	The number of iterations.
clust	Choice of clustering function (character). Defaults to "agnes".
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"
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.

Value

The returned value is a list with two elements:

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

Note

For now, only hierarchical clustering with the agnes function is implemented.

Author(s)

Marijke Van Moerbeke

References

WANG, B., MEZLINI, M. A., DEMIR, F., FIUME, M., TU, Z., BRUDNO, M., HAIBE-KAINS, B., GOLDENBERG, A. (2014). Similarity Network Fusion for aggregating data types on a genomic scale. Nature. 11(3) pp. 333-337. WANG, B., MEZLINI, M. A., DEMIR, F., FIUME, M., TU, Z., BRUDNO, M., HAIBE-KAINS, B., GOLDENBERG, A. (2014). SNFtool: Similarity Network Fusion. R package version 2.2

See Also

SNF,SNFa,SNFb

Examples

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

MCF7_SNFc=SNFc(list(fingerprintMat,targetMat),type="data",distmeasure=c("tanimoto",
"tanimoto"),normalize=FALSE,method=NULL,NN=10,mu=0.5,T=20,clust="agnes",linkage="ward"
,alpha=0.625,StopRange=FALSE)
```

`targetMat` *The target prediction matrix*

Description

A binary data matrix that contains 477 target predictions for the CMAP MCF7 data.

Usage

```
data("targetMat")
```

Format

The format is: num [1:56, 1:477] 0 0 0 0 0 0 0 0 0 0 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:56] "metformin" "phenformin" "phenyl biguanide" "estradiol" ... ..$: chr [1:477] "Arachidonate_15.lipoxygenase" "Estradiol_17.beta.dehydrogenase_2" "Estradiol_17.beta.dehydrogenase_1" "Lanosterol_synthase" ...

`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 = TRUE, Table = TRUE, CompleteSelectionPlot = FALSE,
ClusterPlot=FALSE,cols=NULL,legendposx=0.5,legendposy=2.4,plottype="new",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.
nrclusters	The number of clusters to cut the dendrogram in.
followMaxComps	Logical for plot. Whether to follow the maximum of objects.
followClust	Logical for plot. Whether to follow the specific cluster.
fusionsLog	To be handed to ReorderToReference.
WeightClust	To be handed to ReorderToReference.
names	Optional. Names of the methods.
SelectionPlot	Logical. Should a plot be produced. Depending on followMaxComps and followClust it focuses on the maximum of compounds or a cluster. It will not be indicated to which cluster compounds moved.
Table	Logical. Should a table with the compounds per method and the shared compounds be produced?
CompleteSelectionPlot	Logical. Should the complete distribution of the selection be plotted? This implies that it will be indicated to which cluster compounds will move.
ClusterPlot	Logical. Plot of specific cluster.
cols	The colors used for the different clusters.
legendposx	The x-coordinate of the legend on all plots.
legendposy	The y-coordinate of the legend on all plots.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

Details

The result is provided with extra information as which compounds of the original selection can be found in this cluster and which are extra. Further, plots of the distribution of the compounds can be produced. One plot follows the complete distribution of the cluster while another one focuses on either the maximum number of compounds or a specific cluster, whatever is specified. It are the number of compounds that are plotted and the first element indicated the number of compounds 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 compounds to follow
`nr.clusters`	the number of clusters the selection is divided over
`nr.min.max.together`	the minimum and maximum number of compounds found together
`perc.min.max.together`	minimum and maximum percentage of compounds found together
`AllClusters`	A list with an element per cluster that contains at least one of the compounds 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.

Author(s)

Marijke Van Moerbeke

Examples

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

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

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

Comps=FindCluster(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")
```

`Ultimate` *Function that performs any aggregated data function*

Description

The function `Ultimate` has the ability to perform multiple of the methods listed above simultaneously. The only necessary input are the data matrices and specification of the options. First, clustering is based on each data matrix separately after which the specified integrative analysis methods are conducted. A plot comparing the results is made automatically with ComparePlot. If weights are involved in the method, a comparison plot of the results for these weights is made as well.

Usage

```
Ultimate(List,type=c("data","dist","clusters"),distmeasure,normalize=FALSE,method=NULL,
StopRange=FALSE,NN = 20,mu = 0.5, T = 20, t = 10, r = NULL, nrclusters = NULL,
nrclusterssep = c(7, 7),nrclustersseq = NULL, weight = NULL, Clustweight = 0.5,
clust = "agnes", linkage=c("ward","ward"),alpha=0.625, gap = FALSE, maxK = 50,
 IntClust = c("ADC", "ADECa", "ADECb","ADECc", "WonM", "CECa", "CECb", "CECc",
  "WeightedClust", "WeightedSim", "SNFa", "SNFb", "SNFc"), fusionsLog = TRUE,
 WeightClust= TRUE, PlotCompare = FALSE, cols = NULL, ...)
```

Arguments

`List`	A list of matrices of the same type. It is assumed that the rows are corresponding to the objects.
`type`	Type indicates whether the provided matrices in "List" are either data or distance matrices obtained from the data. If type="dist" the calculation of the distance matrices is skipped and the methods ADC ,ADECa, ADECb, ADECc, CECa, CECb and CECc are not performed. Type should be one of "data" or "dist".
`distmeasure`	A vector of the distance measures to be used on each data matrix.
`normalize`	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization in `Distance`.
`method`	A method of normalization. Should be one of "Quantile","Fisher-Yates", "standardize","Range" or any of the first letters of these names.
`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.
`NN`	The number of neighbours to be used in SNF.
`mu`	A parameter in SNF.
`T`	The number of iterations in SNF.

t	The number of iterations in ADEC and CEC.
r	Optional. The number of features to take for the random sample in ADEC and CEC.
nrclusters	The number of clusters to cut the dendrogram in for ADEC and the plot.
nrclusterssep	Optional. Vector of the number of clusters to cut the dendrogram in of each data source. If NULL, the value of nclusters is used for each.
nrclustersseq	The sequence of number of clusters to cut the dendrogram in for ADECb, CECb and WonM.
weight	The weights to be used in CEC and WeightedClust.
Clustweight	Optional. To be used for the outputs of CEC or WeightedClust. Then only the result of the Clust element is considered.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	A vector with the choice of inter group dissimilarity (character) for each data set.
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. Indicator if gap statistics should be computed. Setting to $FALSE$ will greatly reduce the computation time.
maxK	The maximum number of clusters to be considered during the gap.
IntClust	Specification of the methods to be applied.
fusionsLog	To be handed to MatrixFunction.
WeightClust	To be handed to MatrixFunction.
PlotCompare	Logical. Should the plot over the methods and weight be produced?
cols	Color scheme to be used in the plots.
...	Options to be given to ComparePlot.

Value

The output of `Ultimate` is a list . The first element contains the results of the clustering of the first data source and the last element on the second data source . In between are the results of the integrative methods.

Author(s)

Marijke Van Moerbeke

Examples

```
## Not run:
data(fingerprintMat)
data(targetMat)
data(Colors2)
L=list(fingerprintMat,targetMat)
```

```
MCF7_All=Ultimate(L,type="data",distmeasure=c("tanimoto","tanimoto"),normalize=FALSE,
method=NULL,StopRange=FALSE,NN=20,alpha=0.5,T=20,t=25,r=NULL,nrclusters=7,
nrclustersseq=c(5,25,1),weight=seq(1,0,-0.1),Clustweight=0.5,clust="agnes",
linkage=c("ward","ward"),alpha=0.625,gap=FALSE,IntClust=c("ADC","ADECa","ADECb",
"ADECc","WonM","CECa","CECb","CECc","WeightedClust","WeightedSim",
"SNFa","SNFb","SNFc"),fusionsLog=TRUE,WeightClust=TRUE,PlotCompare=TRUE,
cols=Colors2)

## End(Not run)
```

`WeightedClust` *Weighted clustering*

Description

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=FALSE,method=NULL,
weight = seq(1, 0, -0.1), WeightClust = 0.5, clust="agnes",
linkage = "ward",alpha=0.625,StopRange=FALSE)
```

Arguments

`List`	A list of matrices of the same type. It is assumed the rows are corresponding with the objects.
`type`	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 of "tanimoto", "euclidean", "jaccard","hamming".
`normalize`	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.
`weight`	Optional. A list of different weight combinations for the data sets in List. If NULL, the weights are determined to be rqual 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. 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.
clust	Choice of clustering function (character). Defaults to "agnes".
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"
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.

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 IncidenceComb
Clust	The result for the weight specified in Clustweight

The value has class 'Weighted'

Note

For now, only hierarchical clustering with the agnes function is implemented.

Author(s)

Marijke Van Moerbeke

Examples

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

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

```
,alpha=0.625,StopRange=FALSE)
```

`WeightedSimClust` *Weighted similarity clustering*

Description

The `WeightedSimClust` function performs weighted similarity clustering. The input can be data matrices of which the distance matrices are computed or clustering results where from the distance matrices are extracted. An optimal weight is chosen with the `DetermineWeight_SimClust` function or can be specified by the user. With the found weight the distance matrices are linearly combined and hierarchical clustering is performed.

Usage

```
WeightedSimClust(List, type = c("data", "dist","clusters"), weight = seq(0, 1, 0.01),
 clust = "agnes",linkage=c("ward","flexible"),alpha=0.625, distmeasure = c("euclidean",
"tanimoto"),normalize=FALSE,method=NULL, gap = FALSE, maxK = 50, nrclusters = NULL,
names = c("B", "FP"), AllClusters = FALSE,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`	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".
`weight`	One specific weight to perform clustering on or a list with different weight combinations. If different weight combinations are provided, the function `Chooseweight` is called and an optimal combination is chosen.
`clust`	Choice of clustering function (character). Defaults to "agnes".
`linkage`	A vector with the choice of inter group dissimilarity (character) for each data set.
`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"
`distmeasure`	A character vector with the distance measure for each data matrix. Should be of "tanimoto", "euclidean", "jaccard", "hamming".
`normalize`	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.

gap	Logical. Whether or not to calculate the gap statistic in the clustering on each data matrix separately. Only if type="data".
maxK	The maximal number of clusters to consider in calculating the gap statistic. Only if type="data".
nrclusters	The number of clusters to cut the dendrogram in.
names	The labels to give to the elements in List.
AllClusters	Logical. Whether clustering should be performed for every weight.
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.
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.
location	If plottype is "pdf", a location should be provided in "location" and the figure is saved there.

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 with four elements:

Dist1	The distance matrix of the first data object
Dist2	The distance matrix of the second data object
Weight	The optimal weight
DistW	The weighted distance matrices for the optimal weight
Clust	The resulting clustering

If AllClusters was specified to be TRUE, a sixth element appears containing the clustering results for all weights. The value has class 'WeightedSimClust'

Author(s)

Marijke Van Moerbeke

References

RAVINDRANATH, A. C.,PERUALILA-TAN, N., KASIM, A.,DRAKAKIS, G., LIGGI, S., BREWERTON, S. C.,MASON, D., BODKIN, M. J., EVANS, D. A., BHAGWAT, A. TALLOEN, W., GOHLMANN, H. W. H., QSTAR Consortium, SHKEDY, Z., BENDER, A. (2015). Connecting gene expression data from connectivity map and in silico target predictions for small molecule mechanism-of-action analysis. Mol. BioSyst. Available at: <http://pubs.rsc.org/En/content/ article-landing/2015/mb/c4mb00328d#!divAbstract>

See Also

DetermineWeight_SimClust

Examples

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

L=list(fingerprintMat,targetMat)

MCF7_WeightSim=WeightedSimClust(L,type="data", weight=seq(0,1,0.01),clust="agnes",
linkage=c("flexible","flexible"),alpha=0.625,distmeasure=c("tanimoto","tanimoto"),
normalize=FALSE,method=NULL,gap=FALSE,maxK=50,nrclusters=7,names=c("FP","B"),
AllClusters=FALSE,StopRange=FALSE,plottype="new",location=FALSE)

## End(Not run)
```

WonM *Weighting on Membership*

Description

Weighting on membership is performed with the WonM function. The first step is to compute the appropriate distance matrices for each data source and to use these for hierarchical clustering. This is executed with the agnes function and the ward link. The user may specify a range of values for the number of clusters to cut the resulting dendrograms in. For each value of number of clusters, an incidence matrix is computed and these are added for each data source separately. Eventually, the sums of the incidence matrices are joined together as well, resulting in one consensus matrix. Hierarchical clustering is performed on the consensus matrix to obtain the final clustering result.

Usage

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

Arguments

List	A list of matrices of the same type. It is assumed the rows are corresponding with the objects.
type	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 of "tanimoto", "euclidean", "jaccard","hamming"..
normalize	Logical. Indicates whether to normalize the distance matrices or not. This is recommended if different distance types are used. More details on standardization 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.
nrclusters	A sequence of the number of clusters to cut the dendrogram in.
clust	Choice of clustering function (character). Defaults to "agnes".
linkage	A vector with the choice of inter group dissimilarity (character) for each data set.
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"
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.

Value

The returned value is list with four elements:

DistM	A list with the distance matrix for each data structure
ClustSep	The hierarchical clustering result on each data set
Consensus	The computed consensus matrix over all data sources
Clust	The resulting clustering

Note

For now, only hierarchical clustering with the agnes function is implemented.

Author(s)

Marijke Van Moerbeke

Examples

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

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

Index