Package 'SPARTAAS'

February 22, 2021

Type Package

Title Statistical Methods for Archaeology

Version 1.0.0

Maintainer Arthur Coulon <arthur-coulon@outlook.fr>

Description Statistical pattern recognition and dating using archaeological artefacts assemblages. Package of statistical tools for archaeology.
hclustcompro(perioclust): Bellanger Lise, Coulon Arthur, Husi Philippe (2020, ISBN:978-3-030-60103-4).
seriograph: Bruno Desachy (2004) <doi:10.3406/pica.2004.2396>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.3)

Imports FactoMineR,
grDevices,dplyr,tidyr,ggplot2,plotly,stringr,colorspace,crayon,
shiny,shinydashboard,shinyjs,shinyjqui,fpc,ggdendro,
rstudioapi,htmlwidgets,shinythemes,markdown,explor,shinyWidgets,scatterD3,
sp,ks,rgdal,foreign,maptools,grid,cluster,leaflet,ape,mapview

Suggests knitr, rmarkdown

RoxygenNote 7.1.1

NeedsCompilation no

Author Lise Bellanger [aut],
Philippe Husi [aut],
Arthur Coulon [aut, cre],
Bruno Desachy [ctb],
Benjamin Martineau [ctb]

Repository CRAN

Date/Publication 2021-02-22 11:20:02 UTC

R topics documented:

SPARTAAS-package *SPARTAAS*

Description

Statistical **Pa**ttern **R**ecognition and da**T**ing using **A**rcheological **A**rtefacts assemblage**S**: hclustcompro(perioclust): compromised hierarchical agglomerative clustering method.

Details

Statistical pattern recognition and dating using archaeological artefacts assemblages. Package of statistical tools for archaeology. hclustcompro(perioclust): Bellanger Lise, Coulon Arthur, Husi Philippe (2020, ISBN:978-3-030-60103-4). seriograph: Bruno Desachy (2004) <doi:10.3406/pica.2004.2396>.

Author(s)

NA

Maintainer: Arthur Coulon <arthur-coulon@outlook.fr>

References

Bellanger L., Coulon A., Husi P., 2020 – Perioclust: a new Hierarchical agglomerative clustering method including temporal or spatial ordering constraints. Springer Series, Studies in Classification, Data Analysis, and Knowledge Organization. <doi: 10.1007/978-3-030-60104-1>

Bellanger L., Husi P., Laghzali Y. (2015). Spatial statistic analysis of dating using pottery: an aid to the characterization of cultural areas in West Central France. In : Traviglia A. ed., Across Space and Time, Proceedings of the 41th International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA-2013), Perth (Australie), Amsterdam University Press : 276-282.

adjacency *Dissimilarity matrix base on connectivity information.*

Description

From the data of a network, we build a contiguity matrix. Based on this matrix, we generate a dissmilarity matrix. The matrix contains only 0 or 1, 1 if there is no relationship and 0 if there is a relationship. The network object is a two-column data frame. The first column contains the elements of the network and the second column contains a list of all other elements related to it. The list is a character string consisting of the names of the elements separated by commas (see example).

Usage

```
adjacency(network)
```

Arguments

network	data frame with 2 columns. The first one contains all the elements (nodes) and the second one a string with all the elements related to it (links).

Value

D	Normalized connection dissimilarity matrix

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--  or do  help(data=index)  for the standard data sets.
library(SPARTAAS)

## network stratigraphic data (Network)
network <- data.frame(
  nodes = c("AI09","AI08","AI07","AI06","AI05","AI04","AI03",
  "AI02","AI01","AO05","AO04","AO03","AO02","AO01","APQR03","APQR02","APQR01"),
  edges = c("AI08,AI06","AI07","AI04","AI05","AI01","AI03","AI02","","","AO04","AO03",
  "AO02,AO01","","","APQR02","APQR01","")
)

dissimilarity <- adjacency(network)
dissimilarity
```

arrondi *Return the rounded value*

Description

Always returns the upper value when the next digit is 5.

Usage

```
arrondi(x, acc)
```

Arguments

x	The number to round
acc	The accuracy (number of digits)

Details

Rounding to a negative number of digits means rounding to a power of ten, so for example arrondi(x, digits = -2) rounds to the nearest hundred.

Value

res	value or vector of values rounded

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
library(SPARTAAS)

x1 <- c(15,25,35,45,55)
round(x1,-1)
arrondi(x1,-1)

x2 <- c(-15,-25,-35,-45,-55)
round(x2,-1)
arrondi(x2, -1)

x3 <- 1.125
round(x3,2)
arrondi(x3, 2)

x4 <- seq(-0.55,0.55,0.1)
```

```
data.frame(
  val = x4,
  round = round(x4,1),
  arrondi = arrondi(x4, 1),
  equal = (arrondi(x4, 1) == round(x4,1))
)
```

CAdist *Distance matrix based on correspondence analysis results*

Description

Run an Correspondences analysis on a contingence table then return the matrix distance of the coordinates (you can choose the number of axes to use to build the distance matrix with the nCP parameter).

Usage

```
CAdist(df, nPC = NULL, graph = TRUE)
```

Arguments

df	Data.frame, matrix or table with the data for the correspondence analysis
nPC	Number of principal components to be retained for the construction of the distance matrix. Must be between 1 and the minimum of ncol - 1 and nrow - 1. Could also be "max".
graph	Logical parameter for plot the Correspondences Analysis (axis1, axis2)

Value

dist	The distance matrix

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
library(SPARTAAS)

## contingency table
cont <- data.frame(
```

```
  Cat10 = c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
  Cat20 = c(4,8,0,0,0,0,0,0,0,6,0,0,0,0,0,0),
  Cat30 = c(18,24,986,254,55,181,43,140,154,177,66,1,24,15,0,31,37),
  Cat40.50 = c(17,121,874,248,88,413,91,212,272,507,187,40,332,174,17,288,224),
  Cat60 = c(0,0,1,0,0,4,4,3,0,3,0,0,0,0,0,0,0),
  Cat70 = c(3,1,69,54,10,72,7,33,74,36,16,4,40,5,0,17,13),
  Cat80 = c(4,0,10,0,12,38,2,11,38,26,25,1,18,4,0,25,7),
  Cat100.101 = c(23,4,26,51,31,111,36,47,123,231,106,21,128,77,10,151,114),
  Cat102 = c(0,1,2,2,4,4,13,14,6,6,0,0,12,5,1,17,64),
  Cat110.111.113 = c(0,0,22,1,17,21,12,20,30,82,15,22,94,78,18,108,8),
  Cat120.121 = c(0,0,0,0,0,0,0,0,0,66,0,58,9,0,116,184),
  Cat122 = c(0,0,0,0,0,0,0,0,0,14,0,34,5,0,134,281),
  row.names = c("AI01","AI02","AI03","AI04","AO03","AI05","AO01","AI07","AI08",
  "AO02","AI06","AO04","APQR01","APQR02","AO05","APQR03","AI09")
)

distance <- CAdist(cont, nPC = "max")
distance

## run without printing the plot
distance <- CAdist(cont, nPC = "max", graph=FALSE)
```

`datacancer` *Data set of cancerology.*

Description

Longitude, latitude, number of thyroid cancers. The data concern two departments of France (Loire Atlantique and Vendee) between 1998 and 2012. For reasons of confidentiality they are simulated data.

Usage

```
data("datacancer")
```

Format

List of two objects with 359 observations on the following 3 variables.

$coord (data.frame):

`longitude` a numeric vector: geographical coordinate

`latitude` a numeric vector: geographical coordinate

$var (vector):

`var` a numeric vector: number of thyroid cancers (simulated)

Author(s)

M. Karakachoff (IR CHU - l'institut du Thorax INSERM UMR 1087 - CNRS UMR 6291) Nantes, France

F. Molinie (resp. Loire-Atlantique-Vendee cancer registry - registre-des-cancers) France

Examples

```
library(SPARTAAS)
data(datacancer)
str(datacancer)
head(datacancer$coord)
str(datacancer$var)
```

---

datarcheo *Data set of archeology*

Description

Latitude, longitude, Absolute difference between two dating and the name of the archeological site. Data concern archaeological contexts datation in West Central France based on a large collection of mediaval pottery finds. Two original statistical models are developed to estimate context dates using pottery. The absolute difference is calculated for each context. Data are based on a collective research about medieval ceramic directed by P. Husi ("La céramique médiévale dans la vallée de la Loire moyenne") and from archeo-statistical modelling project CeraR (L. Bellanger & Ph. Husi).

Usage

```
data("datarcheo")
```

Format

List of three objects with 240 observations on the following 4 variables.

$coord (data.frame):

`longitude` a numeric vector: geographical coordinate

`latitude` a numeric vector: geographical coordinate **$var (vector):**

`regionalized_var` a numeric vector: difference between two dating model **$label (vector):**

`noms` a character vector(factor): name of archeological site

Author(s)

P. Husi IR CNRS, UMR CITERES-LAT, CNRS/Tours University, France :

Examples

```
library(SPARTAAS)
data(datarcheo)
str(datarcheo)
head(datarcheo$coord)
str(datarcheo$var)
levels(datarcheo$label)
```

`hclust` *Hierarchical Clustering*

Description

Overload of hclust for dealing with two dissimilarities matrices. Hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it.

Usage

```
hclust(d, method = "complete", members = NULL, d2 = NULL, alpha = NULL)
```

Arguments

`d`	a dissimilarity structure as produced by dist.
`method`	the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).
`members`	NULL or a vector with length size of d. See the 'Details' section.
`d2`	a second dissimilarity structure as produced by dist.
`alpha`	The mixing parameter in order to generate the D_alpha matrix on which the classical hclust method is applied. Formula: D_alpha = alpha * d + (1-alpha) * d2.

Details

Data fusion (alpha optimal value parameter see hclustcompro_select_alpha. It is necessary to define the appropriate proportion for each data source. This is the first sensitive point of the method that the user must consider. A tool is provided to guide his decision.

Value

hclust

Author(s)

The hclust function is based on Fortran code contributed to STATLIB by F. Murtagh.

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (d, method = "complete", members = NULL, d2 = NULL,
    alpha = NULL)
{
    if (!is.null(d2)) {
        if (!length(d) == length(d2)) {
            stop("d and d2 have not the same size.")
        }
        if (is.null(alpha)) {
            sa <- hclustcompro_select_alpha(d, d2, method = method,
                resampling = FALSE)
            alpha <- sa$alpha[1]
        }
        alpha <- as.numeric(alpha)
        if (!(alpha > 0 & alpha < 1)) {
            warning("Alpha must be between 0 and 1.")
            sa <- hclustcompro_select_alpha(d, d2, method = method,
                resampling = FALSE)
            alpha <- sa$alpha[1]
        }
        d <- dist(alpha * d + (1 - alpha) * d2)
    }
    stats::hclust(d, method, members)
  }
```

`hclustcompro` *hclustcompro*

Description

Compromised Hierarchical bottom-up clustering method. The method use two sources of informations. The merging of the two data sources is done by a parameter (alpha) which allows to weight each source. Formula: D_alpha = alpha * D1 + (1-alpha) * D2

Usage

```
hclustcompro(
  D1,
  D2,
  alpha="EstimateAlphaForMe",
  k=NULL,
  title="notitle",
  method="ward.D2"
)
```

Arguments

D1	First dissimilarity matrix (square matrix) or distance matrix. Could be a contingency table (see CAdist). A factorial correspondences analysis is carried out and the distances are used (Chi-square Metric).
D2	Second dissimilarity matrix (square matrix), same size than D1, or distance matrix.
alpha	The mixing parameter in order to generate the D_alpha matrix. Formula: D_alpha = alpha * D1 + (1-alpha) * D2
k	The number of cluster you want.
title	The title to display on the dendrogram plot.
method	The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).

Details

CAH

Data fusion (alpha optimal value parameter see hclustcompro_select_alpha). It is necessary to define the appropriate proportion for each data source. This is the first sensitive point of the method that the user must consider. A tool is provided to guide his decision.

Cut dendrogram

The division into classes, and subclasses, is the second crucial point. It must be done on the basis of knowledge of the study area and some decision support tools such as the cluster silhouette or the calculation of intra cluster variability (WSS: Within Sum of Square). You can use hclustcompro_subdivide in order tu sub-divide a cluster into sub-cluster.

Value

The function returns a list (class: hclustcompro_cl).

D1	First dissimilarity matrix (square matrix)
D2	Second dissimilarity matrix (square matrix)
D_alpha	The matrix use in the CAH result of the mixing of the two matrix (D1 and D2)
alpha	Alpha

tree	An object of class hclust which describes the tree produced by the clustering process (see hclust)
cluster	The vector of cluster of the selected partition
cutree	Plot of the cut dendrogram
call	How you call the function
cont	Original contingency data (if D1 is a contingency table)

Author(s)

The hclust function is based on Fortran code contributed to STATLIB by F. Murtagh.

A. COULON

L. BELLANGER

P. HUSI

Examples

```
library(SPARTAAS)

#network stratigraphic data (Network)
network <- data.frame(
  nodes = c("AI09","AI08","AI07","AI06","AI05","AI04","AI03",
  "AI02","AI01","AO05","AO04","AO03","AO02","AO01","APQR03","APQR02","APQR01"),
  edges = c("AI08,AI06","AI07","AI04","AI05","AI01","AI03","AI02","","","AO04","AO03",
  "AO02,AO01","","","APQR02","APQR01","")
)
#contingency table
cont <- data.frame(
  Cat10 = c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
  Cat20 = c(4,8,0,0,0,0,0,0,0,6,0,0,0,0,0,0),
  Cat30 = c(18,24,986,254,55,181,43,140,154,177,66,1,24,15,0,31,37),
  Cat40.50 = c(17,121,874,248,88,413,91,212,272,507,187,40,332,174,17,288,224),
  Cat60 = c(0,0,1,0,0,4,4,3,0,3,0,0,0,0,0,0,0),
  Cat70 = c(3,1,69,54,10,72,7,33,74,36,16,4,40,5,0,17,13),
  Cat80 = c(4,0,10,0,12,38,2,11,38,26,25,1,18,4,0,25,7),
  Cat100.101 = c(23,4,26,51,31,111,36,47,123,231,106,21,128,77,10,151,114),
  Cat102 = c(0,1,2,2,4,4,13,14,6,6,0,0,12,5,1,17,64),
  Cat110.111.113 = c(0,0,22,1,17,21,12,20,30,82,15,22,94,78,18,108,8),
  Cat120.121 = c(0,0,0,0,0,0,0,0,0,0,66,0,58,9,0,116,184),
  Cat122 = c(0,0,0,0,0,0,0,0,0,0,14,0,34,5,0,134,281),
  row.names = c("AI01","AI02","AI03","AI04","AO03","AI05","AO01","AI07","AI08",
  "AO02","AI06","AO04","APQR01","APQR02","AO05","APQR03","AI09")
)
#obtain the dissimilarities matrices
distance <- CAdist(cont, nPC = 11)
constraint <- adjacency(network)

#You can also run hclustcompro with the dist matrix directly
hclustcompro(D1 = distance, D2 = constraint, alpha = 0.7, k = 4)
```

`hclustcompro_app` *Launch the shiny application.*

Description

see hclustcompro, hclustcompro_select_alpha, seriograph. You can also check the wiki on the application.

Usage

```
hclustcompro_app()
```

Value

No return value

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

library(SPARTAAS)
if(interactive()){
  hclustcompro_app()
}
```

`hclustcompro_detail_resampling`
Resampling process in detail (one curve by set of clone).

Description

Base on a re-sampling process, we generate clone and we check for which alpha the clone and the original object are separeted on the dendrogram (see below). The function show each set of clone curve.

Usage

```
hclustcompro_detail_resampling(D1, D2 = NULL, acc = 2, method = "ward.D2", iter = 5)
```

Arguments

D1	First dissimilarity matrix or contingency table (square matrix). You can replace D1 by a hclustcompro object (Don't use D2 in this case).
D2	Second dissimilarity matrix or network data (square matrix) same size than D1. If D1 is a hclustcompro object D2 is set to NULL.
acc	Number of digits after the comma for the alpha value.
method	The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).
iter	The number of clones cheked for each observation.

Details

Definition of the criterion:

A criterion for choosing alpha IN [0;1] must be determined by balancing the weights between the two information sources in the final classification. To obtain alpha, we define the following criterion:

$$CorCrit_alpha = |Cor(dist_cophenetic, D1) - Cor(dist_cophenetic, D2)|$$

$$equation(1)$$

The CorCrit_alpha criterium in (1) represents the difference in absolute value between two cophenetic correlation (Cophenetic correlation is defined as the correlation between two distances matrices. It is calculated by considering the half distances matrices as vectors. It measures of how faithfully a dendrogram preserves the pairwise distances between the original unmodeled data points). The first correlation is associated with the comparison between D1 and ultrametric distances from the HAC with alpha fixed; while the second compares D2 and ultrametric distances from the HAC with alpha fixed. Then, in order to compromise between the information provided by D1 and D2, we decided to estimate alpha with hat(alpha) such that:

$$hat(alpha) = minCorCrit_alpha$$

$$equation(2)$$

Resampling strategy:

To do this, a set of "clones" is created for each observation i. A clone c of observation i is a copy of observation i for which the adjacency relationships to others have been modified. The clone has none conection exept with j. A set is generated by varying j for all observations except i. A HAC

is then carried out using the combination defined in (1) with D1(c) a (n+1)*(n+1) matrix where the observations i and c are identical and D2(c) a (n+1)*(n+1) matrix where the clone c of i has different neighbourhood relationships from those of i.

Intuitively, by varying alpha between 0 and 1, we will be able to identify when the clone and the initial observation will be separated on the dendrogram. This moment will correspond to the value of alpha above which the weight given to information on the connection between observations contained in D2 has too much impact on the results compared to that of D1.

For a dataset composed of n elements, we will be able to create n*(n-1) clones.

Let CorCrit_alpha(c) defines the same criterion as in (1) in which D1 ans D2 are replaced respectively by D1(c) and D2(c). The estimated alpha is the average of estimated values for each clone. For each clone (c):

$$hat(alpha)(c) = minCorCrit_alpha(c)$$

$$equation(3)$$

hat(alpha)^* is the average of the hat(alpha)(c). In the same spirit as confidence intervals based on bootstrap percentiles (Efron & Tibshirani, 1993), a percentile confidence interval based on replication is also be obtained using the empirical percentiles of the distribution of hat(alpha)(c).

$$hat(alpha)^* = (1/n(n-1)) * sumhat(alpha)(c)$$

$$equation(4)$$

$$cIN[1; n(n-1)].$$

Value

`plot`	The interactive plot: CorCrit_alpha criterium for each resampling dataset

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
###################################
#      For view the equation      #
###################################

plot(
  c(.6,.6,.6,.6),
  c(.9,.5,-.3,-.7),
  xlim = c(.6,1.4),
  ylim = c(-1.1,1),
  axes = FALSE,
  main = "Equations:",
  xlab = "",
  ylab = "",
  pch = 1
```

```
)
text(.65, .9, "( 1 )")
text(.65, .5, "( 2 )")
text(.65,-.3, "( 3 )")
text(.65,-.7, "( 4 )")

text(1, .9,
 expression(CorCrit[alpha] ==  abs(Cor(dist[cophenetic],dist[ceramic]) - Cor(dist[cophenetic],
  dist[stratigraphic])
)))
text(1, .5, expression(hat(alpha) == min(CorCrit[alpha], alpha)))

text(1,-.3, expression(hat(alpha)^(c) == min(CorCrit[alpha]^(c), alpha)))
text(1,-.7, expression(hat(alpha)^"*" == frac(1,n(n-1)) * sum(hat(alpha)^(c),c==1,n(n-1))))

#################################


##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
library(SPARTAAS)

#network stratigraphic data (Network)
network <- data.frame(
  nodes = c("AI09","AI08","AI07","AI06","AI05","AI04","AI03",
  "AI02","AI01","AO05","AO04","AO03","AO02","AO01","APQR03","APQR02","APQR01"),
  edges = c("AI08,AI06","AI07","AI04","AI05","AI01","AI03","AI02","","","AO04","AO03",
  "AO02,AO01","","","APQR02","APQR01","")
)
#contingency table
cont <- data.frame(
  Cat10 = c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
  Cat20 = c(4,8,0,0,0,0,0,0,0,6,0,0,0,0,0,0,0),
  Cat30 = c(18,24,986,254,55,181,43,140,154,177,66,1,24,15,0,31,37),
  Cat40.50 = c(17,121,874,248,88,413,91,212,272,507,187,40,332,174,17,288,224),
  Cat60 = c(0,0,1,0,0,4,4,3,0,3,0,0,0,0,0,0,0),
  Cat70 = c(3,1,69,54,10,72,7,33,74,36,16,4,40,5,0,17,13),
  Cat80 = c(4,0,10,0,12,38,2,11,38,26,25,1,18,4,0,25,7),
  Cat100.101 = c(23,4,26,51,31,111,36,47,123,231,106,21,128,77,10,151,114),
  Cat102 = c(0,1,2,2,4,4,13,14,6,6,0,0,12,5,1,17,64),
  Cat110.111.113 = c(0,0,22,1,17,21,12,20,30,82,15,22,94,78,18,108,8),
  Cat120.121 = c(0,0,0,0,0,0,0,0,0,0,66,0,58,9,0,116,184),
  Cat122 = c(0,0,0,0,0,0,0,0,0,0,14,0,34,5,0,134,281),
  row.names = c("AI01","AI02","AI03","AI04","AO03","AI05","AO01","AI07","AI08",
  "AO02","AI06","AO04","APQR01","APQR02","AO05","APQR03","AI09")
)


hclustcompro_detail_resampling(D1 = CAdist(cont), D2 = adjacency(network))
```

`hclustcompro_select_alpha` *Estimation of the optimal value(s) for the alpha parameter.*

Description

The following criterion "balances"" the weight of D1 and D2 in the final clustering. The alpha value is only a point estimate but the confidence interval gives a range of possible values.

Based on a resampling process, we generate clones and recalculate the criteria according to alpha (see below).

Usage

```
hclustcompro_select_alpha(D1,D2, acc=2, resampling=TRUE, method="ward.D2",iter=5)
```

Arguments

`D1`	First dissimilarity matrix or contingency table (square matrix)
`D2`	Second dissimilarity matrix or network data (square matrix) same size than D1
`acc`	Number of digits after the comma for the alpha value
`resampling`	Logical for estimate the confidence interval with resampling strategy. If you have a lot of data you can save calculation time by setting this option to FALSE
`method`	The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC)
`iter`	The number of clones cheked for each observation. (default: 5 ~ 2mins)

Details

Definition of the criterion:

A criterion for choosing alpha IN [0;1] must be determined by balancing the weights between the two information sources in the final classification. To obtain alpha, we define the following criterion:

$$CorCrit_alpha = |Cor(dist_cophenetic, D1) - Cor(dist_cophenetic, D2)|$$

$$equation(1)$$

The CorCrit_alpha criterium in (1) represents the difference in absolute value between two cophenetic correlation (Cophenetic correlation is defined as the correlation between two distances matrices. It is calculated by considering the half distances matrices as vectors. It measures of how faithfully a dendrogram preserves the pairwise distances between the original unmodeled data points).

The first correlation is associated with the comparison between D1 and ultrametric distances from the HAC with alpha fixed; while the second compares D2 and ultrametric distances from the HAC with alpha fixed. Then, in order to compromise between the information provided by D1 and D2, we decided to estimate alpha with hat(alpha) such that:

$$hat(alpha) = minCorCrit_alpha$$

$$equation(2)$$

Resampling strategy:

To do this, a set of "clones" is created for each observation i. A clone c of observation i is a copy of observation i for which the adjacency relationships to others have been modified. The clone has none conection exept with j. A set is generated by varying j for all observations except i. A HAC is then carried out using the combination defined in (1) with D1(c) a (n+1)X(n+1) matrix where the observations i and c are identical and D2(c) a (n+1)X(n+1) matrix where the clone c of i has different neighbourhood relationships from those of i.

Intuitively, by varying alpha between 0 and 1, we will be able to identify when the clone and the initial observation will be separated on the dendrogram. This moment will correspond to the value of alpha above which the weight given to information on the connection between observations contained in D2 has too much impact on the results compared to that of D1.

For a dataset composed of n elements, we will be able to create n*(n-1) clones.

Let CorCrit_alpha(c) defines the same criterion as in (1) in which D1 ans D2 are replaced respectively by D1(c) and D2(c). The estimated alpha is the average of estimated values for each clone. For each clone (c):

$$hat(alpha)(c) = minCorCrit_alpha(c)$$

$$equation(3)$$

hat(alpha)^* is the average of the hat(alpha)(c). In the same spirit as confidence intervals based on bootstrap percentiles (Efron & Tibshirani, 1993), a percentile confidence interval based on replication is also be obtained using the empirical percentiles of the distribution of hat(alpha)(c).

$$hat(alpha)^* = (1/n(n-1)) * sumhat(alpha)(c)$$

$$equation(4)$$

$$cIN[1; n(n-1)].$$

Warnings:

It is possible to observe an alpha value outside the confidence interval. This problem can be solved, in some cases, by increasing the number of iterations or by changing the number of axes used for the construction of the matrix D1 following the correspondence analysis. If alpha nevertheless remains outside the interval, it means that the data is noisy and the resampling procedure is affected.

Value

The function returns a list (class: selectAlpha_obj).

alpha	The estimate value of the parameter alpha (min CorCrit_alpha)
alpha.plot	The CorCrit for all the possible alpha

If resampling = TRUE

sd	The standard deviation
conf	The confidence interval of alpha.
boxplot	boxplot of alpha estimation with resampling
values	All the potential alpha values obtained from clones

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
###################################
#      For view the equation      #
###################################

plot(
  c(.6,.6,.6,.6),
  c(.9,.5,-.3,-.7),
  xlim = c(.6,1.4),
  ylim = c(-1.1,1),
  axes = FALSE,
  main = "Equations:",
  xlab = "",
  ylab = "",
  pch = 1
)
text(.65, .9, "( 1 )")
text(.65, .5, "( 2 )")
text(.65,-.3, "( 3 )")
text(.65,-.7, "( 4 )")

text(1, .9,
 expression(CorCrit[alpha] ==  abs(Cor(dist[cophenetic],dist[ceramic]) - Cor(dist[cophenetic],
  dist[stratigraphic])
)))
text(1, .5, expression(hat(alpha) == min(CorCrit[alpha], alpha)))

text(1,-.3, expression(hat(alpha)^(c) == min(CorCrit[alpha]^(c), alpha)))
text(1,-.7, expression(hat(alpha)^"*" == frac(1,n(n-1)) * sum(hat(alpha)^(c),c==1,n(n-1))))
```

```
##################################

##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
library(SPARTAAS)

#network stratigraphic data (Network)
network <- data.frame(
  nodes = c("AI09","AI08","AI07","AI06","AI05","AI04","AI03",
  "AI02","AI01","AO05","AO04","AO03","AO02","AO01","APQR03","APQR02","APQR01"),
  edges = c("AI08,AI06","AI07","AI04","AI05","AI01","AI03","AI02","","","AO04","AO03",
  "AO02,AO01","","","APQR02","APQR01","")
)
#contingency table
cont <- data.frame(
  Cat10 = c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
  Cat20 = c(4,8,0,0,0,0,0,0,0,6,0,0,0,0,0,0,0),
  Cat30 = c(18,24,986,254,55,181,43,140,154,177,66,1,24,15,0,31,37),
  Cat40.50 = c(17,121,874,248,88,413,91,212,272,507,187,40,332,174,17,288,224),
  Cat60 = c(0,0,1,0,0,4,4,3,0,3,0,0,0,0,0,0,0),
  Cat70 = c(3,1,69,54,10,72,7,33,74,36,16,4,40,5,0,17,13),
  Cat80 = c(4,0,10,0,12,38,2,11,38,26,25,1,18,4,0,25,7),
  Cat100.101 = c(23,4,26,51,31,111,36,47,123,231,106,21,128,77,10,151,114),
  Cat102 = c(0,1,2,2,4,4,13,14,6,6,0,0,12,5,1,17,64),
  Cat110.111.113 = c(0,0,22,1,17,21,12,20,30,82,15,22,94,78,18,108,8),
  Cat120.121 = c(0,0,0,0,0,0,0,0,0,0,66,0,58,9,0,116,184),
  Cat122 = c(0,0,0,0,0,0,0,0,0,0,14,0,34,5,0,134,281),
  row.names = c("AI01","AI02","AI03","AI04","AO03","AI05","AO01","AI07","AI08",
  "AO02","AI06","AO04","APQR01","APQR02","AO05","APQR03","AI09")
)

dissimilarity <- CAdist(cont,nPC="max",graph=FALSE)
constraint <- adjacency(network)

hclustcompro_select_alpha(D1 = dissimilarity, D2 = constraint)
hclustcompro_select_alpha(D1 = dissimilarity, D2 = constraint, acc = 3, resampling = TRUE)
```

`hclustcompro_subdivide` *subdivide a cluster after runing hclustcompro.*

Description

Allow the user to subdivide one cluster into sub clusters. This function work only with 'hclustcompro_cl' object return by hclustcompro function.

Usage

```
hclustcompro_subdivide(hclustcompro_cl,cluster,nb_class)
```

Arguments

hclustcompro_cl	A hclustcompro_cl object
cluster	The number of the cluster. Numbered from left to right on the dendrogram (1, 2, ...)
nb_class	The number of sub-clusters you want

Value

hclustcompro_cl

A new hclustcompro_cl object updated see hclustcompro

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
library(SPARTAAS)

#network stratigraphic data (Network)
network <- data.frame(
  nodes = c("AI09","AI08","AI07","AI06","AI05","AI04","AI03",
  "AI02","AI01","AO05","AO04","AO03","AO02","AO01","APQR03","APQR02","APQR01"),
  edges = c("AI08,AI06","AI07","AI04","AI05","AI01","AI03","AI02","","","AO04","AO03",
  "AO02,AO01","","","APQR02","APQR01","")
)
#contingency table
cont <- data.frame(
  Cat10 = c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
  Cat20 = c(4,8,0,0,0,0,0,0,0,6,0,0,0,0,0,0),
  Cat30 = c(18,24,986,254,55,181,43,140,154,177,66,1,24,15,0,31,37),
  Cat40.50 = c(17,121,874,248,88,413,91,212,272,507,187,40,332,174,17,288,224),
  Cat60 = c(0,0,1,0,0,4,4,3,0,3,0,0,0,0,0,0,0),
  Cat70 = c(3,1,69,54,10,72,7,33,74,36,16,4,40,5,0,17,13),
  Cat80 = c(4,0,10,0,12,38,2,11,38,26,25,1,18,4,0,25,7),
  Cat100.101 = c(23,4,26,51,31,111,36,47,123,231,106,21,128,77,10,151,114),
  Cat102 = c(0,1,2,2,4,4,13,14,6,6,0,0,12,5,1,17,64),
  Cat110.111.113 = c(0,0,22,1,17,21,12,20,30,82,15,22,94,78,18,108,8),
  Cat120.121 = c(0,0,0,0,0,0,0,0,0,0,66,0,58,9,0,116,184),
  Cat122 = c(0,0,0,0,0,0,0,0,0,0,14,0,34,5,0,134,281),
```

```
  row.names = c("AI01","AI02","AI03","AI04","AO03","AI05","AO01","AI07","AI08",
  "AO02","AI06","AO04","APQR01","APQR02","AO05","APQR03","AI09")
)
#obtain the dissimilarities matrices
distance <- CAdist(cont, nPC = 11)
constraint <- adjacency(network)

#You can also run hclustcompro with the dist matrix directly
clustering <- hclustcompro(D1 = distance, D2 = constraint, alpha = 0.7, k = 7) #number of cluster 7
clustering <- hclustcompro_subdivide(clustering,cluster = 5, nb_class = 2)

#subdivide more than one cluster
clustering2 <- hclustcompro(D1 = distance, D2 = constraint,0.7,k=7) #number of cluster 7
clustering2 <- hclustcompro_subdivide(clustering2,cluster = c(5,7), nb_class = c(2,2))
```

`overlap` *Temporal overlap index*

Description

The overlap index is the ratio between internal overlap and total overlap over time. We define the total overlap limit as follows: the minimum of the lower limits of the pair of individuals and the maximum of the upper limits. We define the limit of the internal overlap as follows: the maximum of the lower limits and the minimum of the upper limits.

Usage

```
overlap(temporal)
```

Arguments

`temporal`	A data frame with tree columns: the name of the element, the lower limit and the upper limit.

Details

The lower and upper limits must be number.

Value

`overlap_matrix`
: The dissimilarity matrix with for each couple i,j the value of the overlap index.

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
library(SPARTAAS)
```

`seriograph` *Plot seriograph (B. DESACHY).*

Description

Visualization of contingency data over time. **Rows** must be individuals (archaeological site,...) and **columns** must be categories (type,...).

Usage

```
seriograph(cont, order, insert, show, permute, col_weight)
```

Arguments

`cont`	Contingency table or hclustcompro object. Note: Your contingency table must have the rows sorted in chronological order. (the order parameter allows you to change the order of the rows if necessary)
`order`	Vector for change the order of the rows (use row's names or cluster names if cont is a hclustcompro object, as a character vector). The oldest one at the bottom. Missing names will not be ploted. You can remove row by simply remove the name in the vector.
`show`	The element to plot. This should be (an unambiguous abbreviation of) one of "both", "EPPM" or "frequency".
`permute`	Logical for permute columns in order to show seriation.
`col_weight`	Logical for activiate or not the coloration of the last column: weight.
`insert`	Vector with the position after where you want insert one or more Hiatus. Could be a list with two vector: position and label to print instead of hiatus. (see last examples)

Details

Seriograph

We chose the seriograph (B. DESACHY). This tool makes it possible to highlight artisanal evolutions over time as well as to understand commercial relations thanks to imported potteries. In this representation, the oldest element is at the bottom. The percentages of each pottery category are displayed. The percentages are calculated independently for each class. The percentage display

allows you to compare the different classes but does not provide information on the number of individuals per class. To fill this gap, the proportion of each class of their workforce is displayed on the seriograph (weight).

We can generalized this representation for other contingency data or with hclustcompro object.

The visualization of positive deviations from the average percentage allows us to observe a series that results from changes in techniques and materials dedicated to ceramic manufacturing over time.

Positive deviation from the average percentage (EPPM in French)

The average percentage is calculated for each ceramic category (columns) on the total number of accounts (all classes combined). From the average percentage we recover for each category and for each rows the difference between the percentage of the category in the class with the average percentage. The EPPM corresponds to the notion of independence deviation (between rows and columns, between categories and time classes) in a chi-square test approach. Although this approach is fundamental in statistical analysis, independence deviations are here purely indicative and are not associated with a p_value that could determine the significance of deviations.

Weight

Weight is the number of observations divided by the total number of observations. It indicates for each row the percentage of the data set used to calculate the frequencies of the elements (row).

Permutation

order argument:
The rows of the contingency table are initially in the order of appearance (from top to bottom). It must be possible to re-order the classes in a temporal way (You can also order as you want your contingency table).

permute argument:
In addition, it is possible to swap ceramic categories (contingency table columns) in order to highlight a serialization phenomenon. Matrix permutation uses an algorithm called "reciprocal averages". Each line is assigned a rank ranging from 1 to n the number of lines. A barycentre is calculated for each column by weighting according to the row rank. Finally, the columns are reorganized by sorting them by their barycentre.

Insert

It's possible to insert a row in the seriograph in order to represent a archeological hiatus or other temporal discontinuities.

Value

The function returns a list (class: seriograph).

`seriograph`	The seriograph plot
`dendrogram`	If cont is a hclustcompro object return the dendrogram with the period order as label
`contingency`	Data frame of the contingencies data group by cluster
`frequency`	Data frame of the frequencies data group by cluster
`ecart`	Data frame of the gap data group by cluster

Author(s)

Desachy Bruno. Le sériographe EPPM : un outil informatisé de sériation graphique pour tableaux de comptages. In: Revue archéologique de Picardie, n°3-4, 2004. Céramiques domestiques et terres cuites architecturales. Actes des journées d'étude d'Amiens (2001-2002-2003) pp. 39-56. DOI : https://doi.org/10.3406/pica.2004.2396.

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
library(SPARTAAS)

## network stratigraphic data (Network)
network <- data.frame(
  nodes = c("AI09","AI08","AI07","AI06","AI05","AI04","AI03",
  "AI02","AI01","AO05","AO04","AO03","AO02","AO01","APQR03","APQR02","APQR01"),
  edges = c("AI08,AI06","AI07","AI04","AI05","AI01","AI03","AI02","","","AO04","AO03",
  "AO02,AO01","","","APQR02","APQR01","")
)
## contingency table
cont <- data.frame(
  Cat10 = c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
  Cat20 = c(4,8,0,0,0,0,0,0,0,6,0,0,0,0,0,0),
  Cat30 = c(18,24,986,254,55,181,43,140,154,177,66,1,24,15,0,31,37),
  Cat40.50 = c(17,121,874,248,88,413,91,212,272,507,187,40,332,174,17,288,224),
  Cat60 = c(0,0,1,0,0,4,4,3,0,3,0,0,0,0,0,0),
  Cat70 = c(3,1,69,54,10,72,7,33,74,36,16,4,40,5,0,17,13),
  Cat80 = c(4,0,10,0,12,38,2,11,38,26,25,1,18,4,0,25,7),
  Cat100.101 = c(23,4,26,51,31,111,36,47,123,231,106,21,128,77,10,151,114),
  Cat102 = c(0,1,2,2,4,4,13,14,6,6,0,0,12,5,1,17,64),
  Cat110.111.113 = c(0,0,22,1,17,21,12,20,30,82,15,22,94,78,18,108,8),
  Cat120.121 = c(0,0,0,0,0,0,0,0,0,0,66,0,58,9,0,116,184),
  Cat122 = c(0,0,0,0,0,0,0,0,0,0,14,0,34,5,0,134,281),
  row.names = c("AI01","AI02","AI03","AI04","AO03","AI05","AO01","AI07","AI08",
  "AO02","AI06","AO04","APQR01","APQR02","AO05","APQR03","AI09")
)
## default
seriograph(cont)

seriograph(cont,show = "EPPM")
seriograph(cont,show = "frequency")

## clustering <- hclustcompro(cont,network,alpha=0.7,k=7) # number of cluster 7
seriograph(clustering)

## change order with cluster name (letters on dendrogram) to sort them in a chronological order
```

```
seriograph(clustering,order = c("C","F","A","G","E","B","D"))

## Don't allow permutation of columns
seriograph(clustering,order = c("C","F","A","G","E","B","D"),permute = FALSE)

## Don't allow coloration
seriograph(clustering,order = c("C","F","A","G","E","B","D"),col_weight = FALSE)

## insert Hiatus (position, 1 -> after first row from bottom: oldest)
seriograph(clustering,order = c("C","F","A","G","E","B","D"),insert = 2)
seriograph(clustering,order = c("C","F","A","G","E","B","D"),insert = c(2,3))

## insert costum label element
insert <- list(
  position = c(2,3),
  label = c("Hiatus.100years","Missing data")
)
seriograph(clustering,order = c("C","F","A","G","E","B","D"),insert = insert)
```

`timerange` *Plot the timerange of observations sorted by cluster.*

Description

Vizualisation of cluster timerange.

Usage

```
timerange(
  data,
  cluster = NULL,
  add = NULL,
  density = NULL,
  color = NULL,
  reorder_color = FALSE
)
```

Arguments

`data`	data.frame (id, inf, sup) for each observation
`cluster`	vector number cluster of observations
`add`	data.frame of information to display on hover.
`density`	vector of the density for each observation.
`color`	vector of the color for each cluster.
`reorder_color`	Logical for reorder the color.

Value

The function returns a list.

plot	The timerange plot.

Author(s)

A. COULON

B. MARTINEAU

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
library(SPARTAAS)
data <- data.frame(
  id = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),
  Lower_bound = c(400,401,401,350,500,460,482,432,399,
    489,750,740,704,700,758,789,802,755,750,820),
  Upper_bound = c(550,689,755,700,780,700,700,699,650,
    850,1100,1100,1010,889,999,999,1050,1002,1000,1100)
)

cluster = c(1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2)

add <- data.frame(
  Site = c("Angers","Angers","Angers","Angers","Angers",
    "Angers","Angers","Angers","Angers","Angers",
    "Blois","Blois","Blois","Blois","Blois",
    "Blois","Blois","Blois","Blois","Blois")
)

timerange(data, cluster, add)

## with sub group (cluster 1 is sub divided in 2: 1.1 and 1.2)
cluster_with_sub = c(1.1,1.1,1.1,1.1,1.1,1.2,1.2,1.2,1.2,1.2,2,2,2,2,2,2,2,2,2,2)

timerange(data, cluster_with_sub, add)

## with density
density <- c(32,34,35,19,9,25,19,29,28,18,10,13,9,10,9,6,3,7,7,1)
timerange(data=data, cluster=cluster, density=density)
```

Index