

Package ‘SPARTAAS’

July 24, 2026

Type Package

Title Statistical Pattern Recognition and daTing using Archaeological Artefacts assemblageS

Version 1.2.7

Date 2026-07-24

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

URL <https://spartaas.gitpages.huma-num.fr/r-package/>

Description Statistical pattern recognition and dating using archaeological artefacts assemblages.

Package of statistical tools for archaeology.

hclustcompro()/perioclust(): Bellanger Lise, Coulon Arthur, Husi Philippe (2021, ISBN:978-3-030-60103-4).

mapclust(): Bel-

langer Lise, Coulon Arthur, Husi Philippe (2021) <doi:10.1016/j.jas.2021.105431>.

seriograph(): Desachy Bruno (2004) <doi:10.3406/pica.2004.2396>.

cerardat(): Bellanger Lise, Husi Philippe (2012) <doi:10.1016/j.jas.2011.06.031>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports FactoMineR,grDevices,dplyr,tidyr,ggplot2,plotly,stringr,colorspace,shiny,shinydashboard,shinyjs,shinyjqui,fpc,ggdendro,htmltools,htmlwidgets,shinythemes,explor,shinyWidgets,scatterD3,ks,grid,cluster,leaflet,ape,caret,MASS,ade4,lmtest,nor1mix,shinycssloaders,scales,fastcluster,nloptr

Suggests knitr,rmarkdown,sf

NeedsCompilation no

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

Repository CRAN

Date/Publication 2026-07-24 10:00:21 UTC

Contents

SPARTAAS-package	2
adjacency	3
arrangePlot	4
arrondi	6
CAdist	8
cerardat	9
cerardat_app	12
cerardat_estim_nf	13
datacancer	15
dataceram	16
datacerardat	17
datangkor	17
datarcheo	18
extract_results	19
hclust	20
hclustcompro	22
hclustcompro_app	23
hclustcompro_cutree	24
hclustcompro_select_alpha	25
mapclust	28
mapclust_app	31
mapclust_cut_tree	32
optim_select_alpha	34
overlap	35
perioclust	36
perioclust_subdivide	38
plot.cerardat_obj	39
seriograph	41
serio_app	44
timerange	45
Index	47

SPARTAAS-package	<i>SPARTAAS</i>
------------------	-----------------

Description

Statistical Pattern Recognition and daTing using Archeological Artefacts assemblageS.

Details

Statistical pattern recognition and dating using archaeological artefacts assemblages. Package of statistical tools for archaeology. hclustcompro()/perioclust(): Bellanger Lise, Coulon Arthur, Husi Philippe (2021, ISBN:978-3-030-60103-4). mapclust(): Bellanger Lise, Coulon Arthur, Husi Philippe (2021) <doi:10.1016/j.jas.2021.105431>. seriograph(): Desachy Bruno (2004) <doi:10.3406/pica.2004.2396>. cerardat(): Bellanger Lise, Husi Philippe (2012) <doi:10.1016/j.jas.2011.06.031>.

Author(s)

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

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

References

Bellanger L., Coulon A., Husi P., 2021 – 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 based on connectivity information.

Description

From the data of a network, we build a contiguity matrix. Based on this matrix, we generate a dissimilarity 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 data frame with two columns. 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 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 contains all elements (nodes) and the second a string with all related elements (links).
---------	---

Value

D	Dissimilarity matrix based on adjacency.
---	--

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
data(datangkor)

## network stratigraphic data (Network)
network <- datangkor$stratigraphy

dissimilarity <- adjacency(network)
dissimilarity
```

arrangePlot	<i>arrangePlot</i>
-------------	--------------------

Description

plot multiple cerardat output

Usage

```
arrangePlot(x,
            which=1:2,
            main="model dateEv (red) and dateAc (black)",
            xlab="Date",
            ylab="Density",
            col1=rgb(0.93,0.23,0.23,0.5),
            col2="black",
            xlim=NULL,
            ylim=NULL,
            ov="topright",
            date=NULL,
            col_date=NULL,
            lty_date=NULL,...
            )
```

Arguments

x	a cerardat output.
which	Vector containing the plots you want (rownames or row number on the contingency table: numeric).
main	an overall title for the plot: see title title .
xlab	a title for the x axis: see title .
ylab	a title for the y axis: see title .
col1	Color of the Event curve (color name, hex code, rgb function {grDevices} for transparency).
col2	Color of the Accumulation curve (color name, hex code, rgb function {grDevices} for transparency).

xlim	Two numeric values, specifying the left limit and the right limit of the scale.
ylim	Two numeric values, specifying the lower limit and the upper limit of the scale.
ov	Keywords indicating where to place the overlap value. This should be (an unambiguous abbreviation of) one of "topright", "topleft", "bottomright", "bottom-left" or "none". For more precise control over the placement, you can set ov to "none" and use text .
date	x coordinate to highlight with a vertical line accross all plots (vector).
col_date	Vector of color for vertical lines (one for each date, could be numeric, hexadecimal, rgb).
lty_date	Vector of line type (see lty in par , don't use lty with arrangePlot) for vertical lines.
...	other parameters to be passed through to plotting functions (bty, lwd, cex.axis see par).

Author(s)

A. COULON
L. BELLANGER
P. HUSI

Examples

```
data("datacerardat")

resultat = cerardat(df = datacerardat$df,
  row.sup = datacerardat$row.sup,
  date = datacerardat$date,
  nf = NULL,
  confidence = 0.95,
  graph = TRUE
)

#Plot the first 5 with their dateEv fit as vertical line
arrangePlot(resultat,
  which = 1:5,
  date = resultat$prediction$Fit_dateEv[1:5]
)
arrangePlot(resultat,
  which = 1:5,
  xlim = c(300, 600),
  date = resultat$prediction$Fit_dateEv[1:5]
)

#Plot using row names and highlighting custom dates
arrangePlot(resultat,
  which = c("Z067", "Z070", "Z074"),
  main = "Neuvy-Le-Roi",
  xlim = c(300, 1300),
  date = c(669, 745, 850, 1115),
```

```

    col_date = c("red", rgb(0.20, 0.40, 0.30), 4, '#5614be')
  )

#You can change the line type (lty) and the color of the curve
arrangePlot(resultat,
  which = c("Z067", "Z070", "Z074"),
  main = "Neuvy-Le-Roi",
  col1 = rgb(0.7, 0.7, 0.7, 0.6), # the last arg of rgb is alpha for opacity
  col2 = "black",
  xlim = c(300, 1300),
  date = c(669, 745, 850, 1115),
  col_date = "grey50",
  lty_date = c(1, 2, 3, 4) # 1: line, 2: dashed line, 3: dotted line, 4:mixed dotted dashed line)
)

#Default first 10 plot
arrangePlot(resultat, which = 1:10)

#You can remove the box arround the plot region
arrangePlot(resultat,
  which = 1:10,
  xlim = c(200, 1000),      # fix the x axis between 200 and 1000
  date = seq(250, 950, 50), # add one line every 50 years between 250 and 950
  col_date = "grey80",
  lty_date = 3,             # dotted line (1: line, 2: dashed line)
  bty = 'n',               # remove box ('L': keep bottom and left)
  cex.axis = 0.8           # control axis text size (only y axis here)
)

#minimalist version (remove y-axis)
arrangePlot(resultat,
  which = 1:10,
  xlim = c(200, 1000),      # fix the x axis between 200 and 1000
  date = seq(250, 950, 50), # add one line every 50 years between 250 and 950
  col_date = "grey80",
  lty_date = 3,             # dotted line (1: line, 2: dashed line)
  ylab = '',               # remove the y-axis name ('Density')
  fg = rgb(0, 0, 0, 0),    # color box transparent (invisible)
  col.axis = rgb(0, 0, 0, 0), # color axis text (only y axis here) transparent (invisible)
  ov = "none"              # remove the overlap percentage
)

```

arrondi

Returns the rounded value.

Description

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

Usage

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

Arguments

x	The number to round.
acc	Accuracy (number of digits). A negative number of digits means rounding to the power of ten, e.g. <code>arrondi(x, digits = -2)</code> will round to the nearest hundred.

Value

res	Value or vector of values rounded.
-----	------------------------------------

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
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

Perform a correspondence analysis on a contingency table and then return the distance matrix 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 plotting the Correspondence Analysis (Axis1, Axis2).

Value

D	The distance matrix
---	---------------------

Author(s)

A. COULON
L. BELLANGER
P. HUSI

Examples

```
data(datangkor)

## contingency table
cont <- datangkor$contingency

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

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

cerardat

cerardat

Description

The methodology is based on a statistical and visual approach using two estimated density curves to date each archaeological context. The statistical procedure required two steps, each leading to the construction of a density curve. The first allowed us to estimate a date corresponding to the terminus post quem of the context, a cursor reflecting an event dated in calendar time. This statistical tool allows the archaeologist to easily visualise and analyse chronological patterns.

Usage

```
cerardat(df, row.sup, date, nf = NULL, confidence = 0.95, graph = T)
```

Arguments

df	The data (data.frame) is a contingency table with the observations in the rows and the technical groups in the columns.
row.sup	Index of supplementary rows in df (vector).
date	The dates of each observation or NA (vector).
nf	an integer representing the number of axes retained in the correspondence analysis. If NULL, it is automatically chosen see cerardat_estim_nf .
confidence	The desired confidence interval (0.95 for 95%).
graph	logical to display the plots or not.

Details

The corpus data is a contingency table with the observations in the rows and the technical groups in the columns. There are two types of observations: the reference corpus observations and the supplementary observations. The supplementary rows (observations) are identified by the argument ‘row.sup’.

step 1: modelling events dated in calendar time (dateEv)

This step involves estimating the date of an event recorded in the ground (an archaeological context for the archaeologist) from the pottery assemblage of which it is composed, by fitting a regression model that relates a known date in calendar time, such as the date of issue of a coin, to its pottery profile. The reference corpus used to fit the regression model. We then used the previously fitted model to calculate a predicted value for contexts not included in the reference corpus, sometimes stratigraphically separated or poorly documented, with a 95% confidence interval for the predicted date.

A correspondence analysis (CA) was carried out to summarize the information in the reference corpus data. We then kept only the first factorial axes. In this way, our contingency table becomes a reduced size table, an incomplete reconstruction of the data. This principle is used in many factor analysis techniques to reduce the number of explanatory variables in the linear regression model.

After estimating the beta parameters of the model using the classical results of multiple regression analysis and checking that the model fits the data correctly, we can deduce the estimated date of an observation and also predict the date of another observation that has no coins and is therefore not dated.

step 2: from event time (dateEv) to accumulation time (dateAc)

We used the results of the first step and the properties of the CA to obtain an estimate of the date of each fabric. We could then define the archaeological time represented as dateAc, in other words the accumulation time of a context, as the weighted sum of the fabric dates; the weights being the proportions of MINVC of each fabric in the context. Assuming that the random variables dateEv_j are independent, the distribution of the accumulation time of each context can be approximated by the Gaussian mixture. In this way, for each context, we obtained a plurimodal density curve representing the estimated law of accumulation time based on the mixture of normal densities (dates of each tissue). By definition, the area under the density curve has a value of 1 (i.e. 100%).

date

In order to estimate a date for the context, it is essential to refer to objects that have been dated by another source, such as coins. These contexts were selected on a very strict basis for their chronos-tratigraphic reliability, level of domestic occupation or enclosures with long urban stratigraphic sequences, thereby minimising any bias associated with the disparity between the date of the coin and that of the context.

overlap

The overlap is calculated as the proportion of the overlapped area over the total area.

See [plot.cerardat_obj](#) and [arrangePlot](#) for visualisation.

Value

prediction	Estimated date for archaeological context (event: dateEV and accumulation: dateAc) with confidence interval. The first two columns show the total count and the number of categories per line (only for columns used in CA). Then a date column shows the known dates. Each dateEv and dateAc model has three columns (value, lower bound, upper bound). The final column show the overlap (%) of the two curves (dateEv and dateAc).
date_gt	Estimated date for technical groups with confidence interval (use for dateAc). The first column show the total count per category in the reference data (only for rows used in CA).
lm	Linear model on the components of the correspondance analysis.
predict_obj_row	date prediction of archaeological contexts (rows) using predict.lm .
predict_obj_col	date prediction of technical groups (columns) using predict.lm .
cont_gt	Contingency table of the reference corpus.
statistical.summary	Statistical summary of the model: Adjusted R-squared R-squared sigma (Residual standard error) The Shapiro-Wilks test is used to verify the normality of the residuals.

	The Durbin-Watson test checks for first order autocorrelation. The Breusch-Pagan test checks for heteroscedasticity.
obs_ca_eval	Quality of row representation in the correspondence analysis.
check_ref	Plot of estimated dates (with confidence interval) and real dates of reference data. Only when the real date is known.
check_sup	Plot of estimated dates (with confidence interval) and real dates of supplementary data. Only when the real date is known.
Shapiro_Wilks	Summary of the Shapiro-Wilks test. see shapiro.test .
Durbin_Watson	Summary of the Durbin-Watson test. see dwtest .
Breusch_Pagan	Summary of the Breusch-Pagan test. see bptest .

Author(s)

A. COULON
L. BELLANGER
P. HUSI

References

Bellanger L. and Husi P. (2012) Statistical tool for dating and interpreting archaeological contexts using pottery. *Journal of Archaeological Science*, Elsevier, 39 (4), pp.777-790. doi:10.1016/j.jas.2011.06.031.

Examples

```
data("datacerardat")

resultat = cerardat(df = datacerardat$df,
  row.sup = datacerardat$row.sup,
  date = datacerardat$date,
  nf = NULL,
  confidence = 0.95,
  graph = TRUE
)

resultat
#The Shapiro-Wilks test is used to verify the normality of the residuals.
#The Durbin-Watson test checks for first order autocorrelation.
#The Breusch-Pagan test checks for heteroscedasticity.

#See the first plot
plot(resultat, which = 1)
#We can add the date (when there is one) and the dateEv and dateAc
abline(v=resultat$prediction$date[1], col="grey40", lty=1)
abline(v=resultat$prediction$Fit_dateEv[1], col="red", lty=2)
abline(v=resultat$prediction$Median_dateAc[1], col="black", lty=3)
legend("topright",
  legend=c("Date", "DateEv", "DateAc"),
  col=c("grey40", "red", "black"),
  lty=c(1, 2, 3))
```

```

)

#You can also create a plot with multiple context (which can be index/integer or rownames)
arrangePlot(resultat,
  which = c("Z067", "Z070", "Z074"),
  main = "Site: Neuvy-Le-Roi.",
  xlim = c(300, 1300),
  date = resultat$prediction[c("Z067","Z070","Z074"),]$Fit_dateEv,
  col_date = c("red", rgb(0.20, 0.40, 0.30), 4)
)

##See the first 10 plots
#plot(resultat, which = 1:10)

#arrange the first 10 plots (with date we can add a grid)
arrangePlot(resultat,
  which = 1:10,
  col1 = rgb(0.7, 0.7, 0.7, 0.6), # the last arg of rgb is alpha for opacity
  col2 = "black",
  xlim = c(200, 1000),          # limit of the x axis between 200 and 1000
  date = seq(250, 950, 50), # add one line every 50 years between 250 and 950
  col_date = "grey80",
  lty_date = 3,                # dotted line
  bty = 'n',                   # remove box
  cex.axis = 0.8               # control axis text size (only y axis is impacted in arrangePlot)
)

##See all the plots
#plot(resultat)

##You can extract the plots and find them in the directory :
#paste0(getwd(),"/figures")
##With the 'extract_results' function
#extract_results(resultat, width=480, height=480, path="figures")

```

cerardat_app

Launch the shiny application.

Description

see [cerardat](#).

Usage

```
cerardat_app()
```

Value

No return value

Author(s)

A. COULON
L. BELLANGER
P. HUSI

Examples

```
if(interactive()){
  cerardat_app()
}
```

cerardat_estim_nf	<i>Number of axes to keep</i>
-------------------	-------------------------------

Description

Estimate the correct number of axes to keep in the regression model (see [cerardat](#)). We use a resampling procedure see [train](#).

Usage

```
cerardat_estim_nf(df, row.sup, date, option=list(method = "LOOCV"))
```

Arguments

df	The data (data.frame) is a contingency table with the observations in rows and the technical groups in columns.
row.sup	Index of supplementary rows in df (vector).
date	The dates of each observation or NA (vector).
option	A list of values that define how the resampling method acts. See detail.

Details

We implement a resampling procedure (see [train](#)) and calculate several metrics in order to determine the number of axes to retain (nf).

The metrics are: PRESS (always use LOOCV for calculations), RMSE, MAE, and adjusted R². In addition, hypothesis tests for normality ([shapiro.test](#)), heteroscedasticity ([bptest](#)) and autocorrelation ([dwtest](#)) are carried out for each value of nf.

You can give a list of options, there are 3 diffrents way of resampling implemented :

Leave-One-Out: 'option=list(method="LOOCV")' (Default)

K-fold cross-validation: 'option=list(method="cv", number=5)'

repeated K-fold cross-validation: 'option=list(method="repeatedcv", number=5, repeats=10)'

Warning: For the CV and repeated CV methods, ensure you have sufficient data per fold. These approaches are particularly useful when the computation time for LOOCV becomes too long (For small datasets, they may generate warnings).

Note: when using the LOOCV method, the values for the Root Mean Square Error (RMSE) are essentially the same as those for PRESS, with the difference that RMSE is the mean rather than the sum.

Value

nf	Number of axes to keep (default minimal PRESS value)
PRESS	plot of the Predicted Residual Error Sum of Squares.
RMSE	plot of the Root Mean Squared Error.
MAE	plot of the Mean Absolute Error.
adj.R_sq	plot of the Coefficient of determination R^2 .
Pvalue	plot of p-values from statistical tests on the hypotheses.
data	data frame of MSE, PRESS and R_sq values.
data_stat	data frame of the p-values.

Author(s)

A. COULON
L. BELLANGER
P. HUSI

References

Bellanger L. and Husi P. (2012) Statistical tool for dating and interpreting archaeological contexts using pottery. *Journal of Archaeological Science*, Elsevier, 39 (4), pp.777-790. doi:10.1016/j.jas.2011.06.031.

Examples

```
data(datacerardat)
res = cerardat_estim_nf(datacerardat$df, datacerardat$row.sup, datacerardat$date)

#Number of axes to keep (minimal PRESS value)
res$nf

#the plots
res$PRESS
res$MAE
res$adj.R_sq

#p-value
res$Pvalue
```

datacancer	<i>Data set of cancerology.</i>
------------	---------------------------------

Description

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

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

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

dataceram	<i>Data set of archeology</i>
-----------	-------------------------------

Description

This important dataset comes from the Collective Research Project (CRP) on Medieval and Modern pottery in the Middle Loire Basin. This is a long-term project, since it began in 1996 and has already been the subject of two books on the subject (Husi dir. 2003 and 2013), as well as an online logical publication (Husi dir. 2022).

Usage

```
data("dataceram")
```

Format

List of three objects with 226 observations.

\$contingency (data.frame) Contingency table of the quantities of 183 types of pottery sherds in the 226 sets.

\$timerange (data.frame) The first column corresponds to the identifier (sets), the second and the third to the lower and upper limits of the temporal range of the observations.

\$geographic_area Vector of the geographical area of each observation.

Author(s)

Husi P. (dir.) – La céramique médiévale et moderne du bassin de la Loire moyenne, chronotypologie et transformation des aires culturelles dans la longue durée (6e —19e s.), XXe Supplément à la Revue Archéologique du Centre de la France, FERACF, Tours, 2022.

Examples

```
data(dataceram)
str(dataceram)
str(dataceram$contingency)
head(dataceram$timerange)
head(dataceram$geographic_area)
```

datacerardat	<i>Data set of archaeology.</i>
--------------	---------------------------------

Description

Data sets on Tours pottery. Contains three objects: - Pottery sherd contingency table (data.frame 280 sets and 391 technical groups) - The date (if known) of the sets (vector) - Column index for supplementary sets (vector)

Usage

```
data("datacerardat")
```

Format

A list with 3 objects.

df Ceramic contingency table. a integer data.frame

date The date (if known) of the sets. a numeric or NA vector

col.sup Column index for supplementary sets. a numeric vector

Examples

```
data(datacerardat)
datacerardat
## maybe str(datacerardat);
```

datangkor	<i>Data set of archeology</i>
-----------	-------------------------------

Description

The archaeological data come from excavations carried out at Angkor Thom (Cambodia), the capital of the Khmer Empire between the 9th and 15th centuries (Gaucher, 2004). The dataset consists of the pottery assemblages (quantities of different types of pottery sherds contained in the sets - `..$contingency`) and the stratigraphy of the sets from 3 disconnected archaeological sites (`..$stratigraphy`).

Usage

```
data("datangkor")
```

Format

List of two objects with 17 observations.

\$contingency (data.frame) Contingency table of the quantities of 12 types of pottery sherds in the 17 sets.

\$stratigraphy (data.frame) Saves the stratigraphic network. The first column corresponds to the nodes (sets) and the second to the edges by listing the nodes connected to it.

Author(s)

Gaucher, J. (2004). Angkor Thom, une utopie réalisée ? Structuration de l'espace et modèle indien d'urbanisme dans le Cambodge ancien. *Arts Asiatiques*, Volume 59, pp. 58-86.

Examples

```
data(datangkor)
str(datangkor)
str(datangkor$contingency)
str(datangkor$stratigraphy)
```

datarcheo

Data set of archeology

Description

Latitude, longitude, absolute difference between two dates and the name of the archaeological site. The data concern the dating of archaeological contexts in West-central France based on a large collection of medieval pottery finds. Two original statistical models are developed to estimate context dates using pottery. The absolute difference is calculated for each context. The data are based on a collective research on medieval pottery directed by P. Husi ("[La céramique médiévale dans la vallée de la Loire moyenne](#)").

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

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

extract_results	<i>Generates all plots (in jpeg format) in a subfolder</i>
-----------------	--

Description

Generates all plots (in jpeg format) in a subfolder (relative path from the working directory).

Usage

```
extract_results(cerardat, width=480, height=480, path="figures",
               main="model dateEv (red) and dateAc (black)",
               xlab="Date", ylab="Density",
               col1 = rgb(0.93,0.23,0.23,0.5), col2 = "black",
               xlim=NULL, ylim=NULL)
```

Arguments

cerardat	a cerardat output.
width	the width of the graphics region in pixels.
height	the width of the graphics region in pixels.
path	directory where plots are exported (relative path from the working directory). Automatically generates two sub-folders 'ref' for the reference plot and 'sup' for the supplementary plot.
main	an overall title for the plot: see title .
xlab	a title for the x axis: see title .
ylab	a title for the y axis: see title .
col1	Color of the the Event curve (color name, hex code, rgb function {grDevices} for transparency).
col2	Color of the the Accumulation curve (color name, hex code, rgb function {grDevices} for transparency).
xlim	Two numeric values, specifying the left limit and the right limit of the scale.
ylim	Two numeric values, specifying the lower limit and the upper limit of the scale.

Author(s)

A. COULON
 L. BELLANGER
 P. HUSI

Examples

```
data("datacerardat")

resultat = cerardat(df = datacerardat$df,
  row.sup = datacerardat$row.sup,
  date = datacerardat$date,
  nf = NULL,
  confidence = 0.95,
  graph = TRUE
)
#Generates all plots (in jpeg format) in a folder 'figures'
#You will find the jpeg in 'ref' and 'sup' subfolder
#extract_results(resultat,width=480, height=480, path="figures",
#  col1=rgb(0.93,0.23,0.23,0.5),
#  col2="black",
#  xlim=NULL,
#  ylim=c(0,0.03)
#  )

#You can extract the plots and find them in the directory :
paste0(getwd(),"/figures")
```

hclust

Hierarchical clustering for multiple datasets (Compromised clustering).

Description

Implementation of [stats::hclust](#) for dealing with multiple dissimilarities matrices. We use [fastcluster::hclust](#), another implementation of hclust with faster algorithms.

Usage

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

Arguments

d	a dissimilarity structure as produced by dist or a list of dissimilarity structure.
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 [stats::hclust](#) ‘Details’ section.

weight The parameter in order to generate a mixed matrix on which the classical hclust method is applied (same length as d).
Formula:

$$D_{\alpha} = w_1 D_1 + w_2 D_2 + \dots$$

where w_1 is the first value of vector weight, D_1 the first matrix of d and D_{α} the mixed matrix used in hclust. Sum of weight must be 1.

Details

Data fusion (parameter weighth): it is necessary to define the appropriate proportion for each data source. This is the first sensitive point of the method that the user has to consider. A tool is provided to help him in his decision see [hclustcompro_select_alpha](#) (case with only 2 matrices) or [optim_select_alpha](#).

Value

See [stats::hclust](#) or [fastcluster::hclust](#)

Author(s)

The stats::hclust function is based on Fortran code contributed to STATLIB by F. Murtagh. The fastcluster::hclust function implements hierarchical clustering with the same interface as hclust from the stats package but with much faster algorithms.

A. COULON

L. BELLANGER

P. HUSI

Examples

```
data(datangkor)

#network stratigraphic data (Network)
network <- datangkor$stratigraphy

#contingency table
cont <- datangkor$contingency

#obtain the dissimilarities matrices
distance <- CAdist(cont, nPC = 11)
constraint <- adjacency(network)

D1 = distance
D2 = constraint

tree = hclust(d = list(D1, D2), method = "ward.D2", weight = c(0.7,0.3))
plot(tree, hang=-1)
```

hclustcompro

*hclustcompro***Description**

Compromised Hierarchical bottom-up clustering method. The method uses at least two sources of information. The merging of the data sources is done by weight that allows to weight each source. Exemple for two sources:

$$D_{\alpha} = \alpha D_1 + (1 - \alpha) D_2$$

Usage

```
hclustcompro(d, method = "complete", weight = NULL)
```

Arguments

d	a list of 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).
weight	The parameter in order to generate a mixed matrix on which the classical hclust method is applied (same length as d). Formula:

$$D_{\alpha} = w_1 D_1 + w_2 D_2 + \dots$$

where w_1 is the first value of vector weight, D_1 the first matrix of d and D_{α} the mixed matrix used in hclust. Sum of weight must be 1.

Details**CAH**

Data fusion (parameter optimal value 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 has to consider. A tool is provided to help him in his decision.

Cut dendrogram

The division into classes (see [hclustcompro_cutree](#)) is the second crucial point. It has to be done based on the knowledge of the study area and some decision support tools such as the cluster silhouette or the calculation of the intra-cluster variability (WSS: Within Sum of Square).

Value

See [stats::hclust](#) or [fastcluster::hclust](#)

Author(s)

A. COULON
L. BELLANGER
P. HUSI

Examples

```
data(datangkor)

#network stratigraphic data (Network)
network <- datangkor$stratigraphy

#contingency table
cont <- datangkor$contingency

#obtain the dissimilarities matrices
distance <- CAdist(cont, nPC = 11)
constraint <- adjacency(network)

#You can also run hclustcompro with the dist matrix directly
tree = hclustcompro(d = list(distance, constraint), weight = c(0.7, 0.3), method = "ward.D2")
cluster = hclustcompro_cutree(tree, k=5, reorder=TRUE, letter=TRUE)
```

hclustcompro_app	<i>Launch the shiny application.</i>
------------------	--------------------------------------

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

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

hclustcompro_cutree	<i>Cut a Tree into Groups of Data.</i>
---------------------	--

Description

Cuts the tree and draws rectangles around the branches of the dendrogram highlighting the corresponding clusters. This function is a combination of [cutree](#) and [rect.hclust](#).

Usage

```
hclustcompro_cutree(tree, k, main="", sub="", xlab="", ylab="",
  reorder=FALSE, letter=FALSE
)
```

Arguments

tree	a tree as produced by hclust. hclustcompro_cutree() only expects a list with components merge, height, and labels, of appropriate content each.
k	an integer scalar or vector with the desired number of groups
main	an overall title for the plot: see title .
sub	a subtitle for the plot: see title .
xlab	a title for the x axis: see title .
ylab	a title for the y axis: see title .
reorder	logical, reorder the number cluster (from left to right on the dendrogram).
letter	logical, replace number by letter (1->A; 2->B...).

Value

hclustcompro_cutree returns a vector with group memberships.

Author(s)

A. COULON
 L. BELLANGER
 P. HUSI

Examples

```

data(datangkor)

#network stratigraphic data (Network)
network <- datangkor$stratigraphy

#contingency table
cont <- datangkor$contingency

#obtain the dissimilarities matrices
distance <- CAdist(cont, nPC = 11)
constraint <- adjacency(network)

#You can also run hclustcompro with the dist matrix directly (number of cluster 7)
tree <- hclustcompro(d = list(distance, constraint), weight = 0.7, method="ward.D2")
clustering <- hclustcompro_cutree(tree, k = 5)

```

hclustcompro_select_alpha

Estimate the optimal value(s) of the α parameter.

Description

The following criterion "balances" the weight of D_1 and D_2 in the final clustering. The α 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 α (see below).

Usage

```

hclustcompro_select_alpha(
  D1,
  D2,
  acc=2,
  method="complete",
  resampling=TRUE,
  iter=5,
  suppl_plot=TRUE
)

```

Arguments

D1	First dissimilarity matrix or contingency table (square matrix).
D2	Second dissimilarity matrix or network data (square matrix) of the same size as D1.

acc	Number of digits after the decimal point 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).
resampling	Logical for estimating the confidence interval with a resampling strategy. If you have a lot of data, you can save computation time by setting this option to FALSE.
iter	The number of clones checked per observation. (200 observations iter=1: ~30 sec, 1000 observations iter=1: ~40 min).
suppl_plot	Logical defines whether additional plots should be displayed.

Details

Definition of the criterion:

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

$$CorCrit_{\alpha} = |Cor(dist_{cophenetic}, D_1) - Cor(dist_{cophenetic}, D_2)|$$

Equation(1)

The $CorCrit_{\alpha}$ criterion in (1) represents the difference in absolute value between two cophenetic correlations (cophenetic correlation is defined as the correlation between two distance matrices. It is calculated by considering the half distance matrices as vectors. It measures how faithfully a dendrogram preserves the pairwise distances between the original unmodeled data points). The first correlation is related to the comparison between D_1 and the ultrametric distances of the clustering with α fixed, while the second compares D_2 and the ultrametric distances of the clustering with α fixed. Then, in order to compromise between the information provided by D_1 and D_2 , we decided to estimate α with $\hat{\alpha}$ such that:

$$\hat{\alpha} = minCorCrit_{\alpha}$$

Equation(2)

Resampling strategy:

This is done by creating a set of "clones" for each observation i . A clone c of observation i is a copy of observation i for which the distances from the second source have been modified. The modification is made by copying the distances for the second source from another observation j . A clustering is then performed using the combination defined in (1) with $D_1^{(c)}$ an $(n+1) \times (n+1)$ matrix where observations i and c are identical and $D_2^{(c)}$ an $(n+1) \times (n+1)$ matrix where the clone c of i has different distances from those of i . A set of clones is generated by varying j for all observations except i . We can generate a set of $n-1$ clones for each element i in n , so $n(n-1)$ clones in total.

Intuitively, by varying α between 0 and 1, we will be able to identify when the clone and the original observation are separated on the dendrogram. This moment will correspond to the value of alpha above which the weight given to the information about the connection between observations contained in D_2 has too much influence on the results compared to that of D_1 .

Let $CorCrit_\alpha^{(c)}$ define the same criterion as in (1), where D_1 and D_2 are replaced by $D_1^{(c)}$ and $D_2^{(c)}$ respectively. The estimated α is the mean of the estimated values for each clone.

For each clone c :

$$\hat{\alpha}^{(c)} = \min CorCrit_\alpha^{(c)}$$

Equation(3)

$\hat{\alpha}^*$ is the mean of $\hat{\alpha}^{(c)}$. 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 $\hat{\alpha}^{(c)}$.

$$\hat{\alpha}^* = \frac{1}{n(n-1)} \sum \hat{\alpha}^{(c)}$$

Equation(4)

$$c \in [1; n(n-1)]$$

Warnings:

It is possible to observe an α value outside the confidence interval. In some cases, this problem can be solved by increasing the number of iterations or by changing the number of axes used to construct the matrix D1 after the correspondence analysis. If the α value remains outside the interval, it means that the data are noisy and the resampling procedure is affected.

Value

The function returns a list (class: selectAlpha_obj).

alpha	The estimated value of the alpha parameter (min CorCrit_alpha)
alpha.plot	The CorCrit curve for alpha between 0 and 1

If resampling = TRUE

sd	The standard deviation
conf	The confidence interval of alpha
boxplot	The boxplot of alpha estimation with resampling
values	All potential alpha values obtained from each clone

Author(s)

A. COULON
L. BELLANGER
P. HUSI

Examples

```
data(datangkor)

#network stratigraphic data (Network)
network <- datangkor$stratigraphy

#contingency table
cont <- datangkor$contingency

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

hclustcompro_select_alpha(D1 = dissimilarity, D2 = constraint, method = "ward.D2")
hclustcompro_select_alpha(D1 = dissimilarity, D2 = constraint, acc = 3,
                          method = "ward.D2", resampling = TRUE)
```

mapclust

Divise hierarchical Clustering using Spatialpatches algorithm.

Description

This function performs a divisive hierarchical clustering on a regionalised variable using the spatial patches algorithm (Woillez et al. 2007; Woillez, Rivoirard and Petitgas 2009). It is a top-down hierarchical clustering with a geographical constraint. It is possible to cut the tree by clicking on the dendrogram at the desired level. The results include a description of the clusters and graphics. When slicing the dendrogram, you can look at the two plots (WSSPlot and AveSilPlot) that show the relatively good quality of the partitions. The first shows the Within Sum of Square (WSS) for each partition and you can use the elbow approach to select a partition. The second graph shows the average silhouette width. This index is between -1 and 1. The closer it is to 1, the better the partition. See [silhouette](#).

If you want to cut the dendrogram to a different dlim or number of clusters, you can do so without re-running mapclust() with [mapclust_cut_tree](#).

Usage

```
mapclust(
  coord,
  var,
  label = "Nolabel",
  iter = 20,
  Plabel = TRUE,
  lonlat = TRUE,
  positive_var = FALSE,
  n = NULL
)
```

Arguments

coord	The x (longitude) and y (latitude) coordinates of the data.frame or matrix dimension 2.
var	The regionalised variable(s) of interest
label	(optional) The names of the samples or an identifier. Must be a factor.
iter	The number of iterations. The number of different dlim you want to test (must be greater than 10).
Plabel	Logical parameter to activate or not the printing of labels on the dendrogram.
lonlat	Logical parameter to activate or not the cartography in lonlat system with leaflet (base map).
positive_var	logical parameter that defines whether your variable of interest is positive or not.
n	Number of groups. If NULL, you can select the number of groups by clicking on the dendrogram.

Details

Dlim is the selected minimum distance from the sample to the patch centre: to identify patches (units are those of coordinates). The dlim is automatically initialised and does not need to be set by the user. The minimum data is a data frame or matrix with at least 3 columns.

Value

the function returns a list.

Plot:

dendrogram	The global dendrogram (hclust object)
dendrogram_ggplot	The global dendrogram (ggplot2 object)
cuttree	The cut dendrogram
map	The map of the selected partition
AveSilPlot	The average silhouette width plot (for each partition)
WSSPlot	The Within Sum of Square plot (for each partition)
silhouette	The silhouette plot of the selected partition

Value:

X	The x-coordinate data you have used
Y	The y-coordinate data you have used
var	The regionalized variable(s) data you have used
label	The label vector you have used
density	The estimated density based on var. Equal to var if you used a unidimensional density variable.
cluster	The cluster vector of the selected partition

Plabel	Logical parameter to activate or not the printing of labels on the dendrogram
fullhist	The compositional cluster for each observation
hist	The compositional cluster without duplicates (corresponds to the split on the dendrogram)
dlim	The vector of the different limit distances
cutdlim	The select dlim for the cut of the current partition
DiMatrix	The weighted euclidean distance matrix
silhouetteData	The silhouette data of the selected partition
AveSilData	The average silhouette value for each partition
Moran	The Moran index for each group for each partition
lonlat	Logical parameter, whether your coordinates are in latitude-longitude format or not

Author(s)

A. COULON L. BELLANGER P. HUSI

References

Bellanger L., Coulon A. and Husi P. (2021) Determination of cultural areas based on medieval pottery using an original divisive hierarchical clustering method with geographical constraint (Map-Clust), Journal of Archaeological Science, Volume 132 doi:[10.1016/j.jas.2021.105431](https://doi.org/10.1016/j.jas.2021.105431).

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.

Wuillez M., Poulard J.C., Rivoirard J., Petitgas P., Bez N. (2007). Indices for capturing spatial patterns and their evolution in time, with application to European hake (*Merluccius merluccius*) in the Bay of Biscay. ICES J. Mar. Sci. 64, 537-550.

Wuillez M., Rivoirard J. and Petitgas P. (2009) Notes on survey-based spatial indicators for monitoring fish populations, Aquatic Living Resources, 22 :155-164.

Examples

```
#####
## loading data
data(datarcheo)
data(datacancer)

#####
### Example: 1
## Function "mapclust"
# object <- mapclust( coord = ..., var = ..., label = ...)

classification <- mapclust(datarcheo$coord, datarcheo$var, datarcheo$label, n=4)
```

```

#Global dendrogram
  classification$dendrogram
#Cut dendrogram
  classification$cuttree
#silhouette of selected partition
  classification$silhouette

#You can cut the dendrogram for another dlim
  NewCut <- mapclust_cut_tree(classification, dlim=0.30)

#See evaluation using Silhouette width by running:
  NewCut$silhouette
#If the plot is empty try to increase the height of the window (full screen)

#See summary of the data by running:
  summary(NewCut$silhouetteData)

#####
## kmeans comparison
# prepare data (only geographical data)
  datakmeans <- datarcheo$coord

#kmeans
  number_cluster <- 4
  cl <- kmeans(datakmeans, number_cluster)
  plot(datakmeans, col = cl$cluster)

```

mapclust_app

Shiny application for Mapclust method

Description

see [mapclust](#) You can also check the wiki on the application.

Usage

```
mapclust_app()
```

Value

No return value

Author(s)

A. COULON
L. BELLANGER
P. HUSI

Examples

```
## open application

if(interactive()){
  mapclust_app()
}
```

mapclust_cut_tree	<i>Function to cut the dendrogram for a new height (distance limit) or a new number of clusters and map the new partition</i>
-------------------	---

Description

The function returns the new map, one dendrogram with the cut line, the silhouette width and the new vector cluster. You must call [mapclust](#) beforehand to get a `mapclust_cl` object that can be used by `mapclust_cut_tree`.

Usage

```
mapclust_cut_tree(classification, nb_grp = NA, dlim = NA)
```

Arguments

<code>classification</code>	The object return by <code>mapclust</code> Class: <code>mapclust_cl</code>
<code>nb_grp</code>	The number of groups you want to have on the partition. Must be an integer. (don't use <code>dlim</code> in this case)
<code>dlim</code>	The value of <code>dlim</code> at which you wish to cut the dendrogram. You can enter any value (numerical) and the function will select the nearest lower <code>dlim</code> with the same partition. (do not use <code>nb_grp</code> in this case).

Value

the function returns a list.

Plot:

<code>dendrogram</code>	The global dendrogram (hclust object)
<code>dendrogram_ggplot</code>	The global dendrogram (ggplot2 object)

cuttree	The cut dendrogram
map	The map of the selected partition
AveSilPlot	The average silhouette width plot (for each partition)
WSSPlot	The Within Sum of Square plot (for each partition)
silhouette	The silhouette plot of the selected partition
Value:	
X	The x-coordinate data you have used
Y	The y-coordinate data you have used
var	The regionalised variable data you have used
label	The label vector you have used
density	The estimated density based on var. Equal to var if you used a unidimensional density variable
cluster	The cluster vector of the selected partition
Plabel	Logical parameter to activate or not the printing of labels on the dendrogram
fullhist	The compositional cluster for each observation
hist	The compositional cluster without duplicates (corresponds to the split on the dendrogram)
dlim	The vector of the different limit distances
cutdlim	The select dlim for the cut of the current partition
DiMatrix	The weighted euclidean distance matrix
silhouetteData	The silhouette data of the selected partition
AveSilData	The average silhouette value for each partition
Moran	The Moran index for each group for each partition
lonlat	Logical parameter, whether your coordinates are in latitude-longitude format or not

Author(s)

A. COULON L. BELLANGER P. HUSI

Examples

```
## loading data
data(datarcheo)

##First you need to run the mapclust function.
#This function allow you to obtain one partition
# object <- mapclust( coord = ..., var = ..., label = ...)
OldClassif <- mapclust(datarcheo$coord, datarcheo$var, datarcheo$label, n = 4)

##In order to cut the dendrogram for another dlim
NewCut <- mapclust_cut_tree(classification = OldClassif, dlim = 0.37)
##In order to cut the dendrogram for another number of cluster
```

```

NewCut2 <- mapclust_cut_tree(classification = OldClassif, nb_grp = 4)

#See evaluation using Silhouette width by running:
NewCut$silhouette
#If the plot is empty try to increase the height of the window (full screen).

#See summary of the data by running:
summary(NewCut$silhouetteData)

```

optim_select_alpha	<i>Optimized selection of the weights for the sources (hclustcompro)</i>
--------------------	--

Description

Description of the function.

Usage

```

optim_select_alpha(data = list(),
                   acc = 2,
                   method = "complete",
                   algo = "NLOPT_GN_ORIG_DIRECT_L"
)

```

Arguments

data	liste des matrices de distances
acc	nombre de chiffre après la virgule
method	stratégie d'aggrégation utilisée par hclustcompro
algo	Algorithme utilisé par nloptr pour trouver les poids optimaux

Details

See NLOPT website

Value

solution	vector of weigths
res_nlopt	object return by nloptr

Author(s)

A. COULON
 L. BELLANGER
 P. HUSI

Examples

```
## The function is currently defined as
```

overlap	<i>Temporal overlap index</i>
---------	-------------------------------

Description

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

The dissimilarity between time periods is calculated as the ratio of the overlap of the time periods (distance in the case of disjoint time periods) to the cumulative extent of the two time periods.

As the ratio is bounded between -1 and 1, we add 1 to make it positive and normalise it so that it is between 0 and 1.

This overlap index then needs to be transformed into a dissimilarity index between sets. To do this we use the 1 - ratio. It is equal to 0 if the two time periods are identical and 1 if they are infinitely different.

Value

D	The dissimilarity matrix base on the overlap index.
---	---

Author(s)

A. COULON
L. BELLANGER
P. HUSI

Examples

```
data(dataceram)
dist <- overlap(dataceram$timerange)
```

perioclust

perioclust

Description

Compromised Hierarchical bottom-up clustering method. The method uses two sources of information. The merging of the two data sources is done by a parameter (α) that allows to weight each source.

$$D_{\alpha} = \alpha D_1 + (1 - \alpha) D_2$$

Usage

```
perioclust(
  D1,
  D2,
  alpha="EstimateAlphaForMe",
  k=NULL,
  title="notitle",
  method="complete",
  suppl_plot=TRUE
)
```

Arguments

D1	First dissimilarity matrix (square matrix) or distance matrix. Could be a contingency table (see CADist). A factorial correspondence analysis is performed using the distances (chi-square metric).
D2	Second dissimilarity matrix (square matrix), same size as D1, or distance matrix.
alpha	The mixing parameter in order to generate the D_{α} matrix (in [0;1]). Formula: $D_{\alpha} = \alpha D_1 + (1 - \alpha) D_2$
k	The number of clusters you want.
title	The title to be displayed 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).
suppl_plot	Logical defines whether additional plots are to be displayed (WSS and average sil plot).

Details

CAH

Data fusion (parameter α optimal value 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 has to consider. A tool is provided to help him in his decision.

Cut dendrogram

The division into classes and subclasses is the second crucial point. It has to be done based on the knowledge of the study area and some decision support tools such as the cluster silhouette or the calculation of the intra-cluster variability (WSS: Within Sum of Square). You can use [perioclust_subdivide](#) to subdivide a cluster into sub-clusters.

Value

The function returns a list (class: `perioclust_cl`):

D1	First dissimilarity matrix (square matrix)
D2	Second dissimilarity matrix (square matrix)
D_alpha	The matrix use in the CAH resulting from the mixing of the two matrices (D1 and D2)
alpha	Alpha
tree	An object of class <code>hclust</code> , describing the tree generated by the clustering process (see hclust)
cluster	The cluster number vector 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)

A. COULON
L. BELLANGER
P. HUSI

Examples

```
data(datangkor)

#network stratigraphic data (Network)
network <- datangkor$stratigraphy

#contingency table
cont <- datangkor$contingency

#obtain the dissimilarities matrices
distance <- CAdist(cont, nPC = 11)
constraint <- adjacency(network)
```

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

perioclust_subdivide *Subdivide a cluster after running perioclust.*

Description

Allow the user to split one cluster into sub-clusters. This function only works with 'perioclust_cl' object returned by the perioclust function.

Usage

```
perioclust_subdivide(perioclust_cl, cluster, nb_class)
```

Arguments

perioclust_cl	A perioclust_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

see [perioclust](#)

Author(s)

A. COULON
L. BELLANGER
P. HUSI

Examples

```
data(datangkor)

#network stratigraphic data (Network)
network <- datangkor$stratigraphy

#contingency table
cont <- datangkor$contingency

#obtain the dissimilarities matrices
distance <- CAdist(cont, nPC = 11)
constraint <- adjacency(network)

#You can also run hclustcompro with the dist matrix directly (number of cluster 7)
```

```

clustering <- perioclust(D1 = distance, D2 = constraint, alpha = 0.7,k=7,method="ward.D2")
clustering <- perioclust_subdivide(clustering,cluster = 5, nb_class = 2)

#subdivide more than one cluster (number of cluster 7)
clustering2 <- perioclust(D1 = distance, D2 = constraint,0.7,k=7, method = "ward.D2")
clustering2 <- perioclust_subdivide(clustering2,cluster = c(5,7), nb_class = c(2,2))

```

plot.cerardat_obj *plot cerardat model*

Description

plot cerardat model

Usage

```

## S3 method for class 'cerardat_obj'
plot(x,
     which = NULL,
     main="model dateEv (red) and dateAc (black)",
     xlab = "Date",
     ylab = "Density",
     col1 = rgb(0.93,0.23,0.23,0.5),
     col2 = "black",
     xlim=NULL,
     ylim=NULL,
     ov="topright",...)

```

Arguments

x	a cerardat output.
which	Vector containing the plots you want (rownames or row number on the contingency table: numeric).
main	an overall title for the plot: see title .
xlab	a title for the x axis: see title .
ylab	a title for the y axis: see title .
col1	Color of the Event curve (color name, hex code, rgb function {grDevices} for transparency).
col2	Color of the Accumulation curve (color name, hex code, rgb function {grDevices} for transparency).
xlim	Two numeric values, specifying the left limit and the right limit of the scale.
ylim	Two numeric values, specifying the lower limit and the upper limit of the scale.
ov	Keywords indicating where to place the overlap value. This should be (an unambiguous abbreviation of) one of "topright", "topleft", "bottomright", "bottomleft" or "none". For more precise control over the placement, you can set ov to "none" and use text .
...	other parameters to be passed through to plotting functions (bty, cex.axis, las).

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```

data("datacerardat")

resultat = cerardat(df = datacerardat$df,
  row.sup = datacerardat$row.sup,
  date = datacerardat$date,
  nf = NULL,
  confidence = 0.95,
  graph = TRUE
)

#See the first plot
plot(resultat, which = 1)
plot(resultat,
  which = 1,
  xlim = c(200, 700),
  ylim = c(0, 0.03),
  col1 = rgb(0.7, 0.7, 0.7, 0.6),
  col2 = "black",
  las = 1,
  ov = "bottomright"
)
abline(v = resultat$prediction$date[1], col="grey60", lty=1)
abline(v = resultat$prediction$Fit_dateEv[1], col="grey60", lty=2)
abline(v = resultat$prediction$Median_dateAc[1], col="grey60", lty=3)
legend("topright",
  legend=c("Date coin", "DateEv", "DateAc"),
  col="grey60",
  lty=c(1, 2, 3)
)

plot(resultat, which = 1, ov = "none")
text(x=600,y=0.01,labels=paste0("overlap: ",resultat$prediction$overlap,"%"),cex=0.8,col="grey20")

#See the first 10 plots
#plot(resultat, which = 1:10)

#See all plots
#plot(resultat)

```

seriograph	<i>Plot seriograph (B. DESACHY).</i>
------------	--------------------------------------

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)
```

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 to change the order of the rows (use row names or cluster names if cont is a hclustcompro object, as a character vector). The oldest (at the bottom) must be at the end of the vector. Missing names are not plotted. You can remove a row by simply removing the name in the vector.
show	The element to be plotted. This should be (a unique abbreviation of) one of 'both', 'EPPM' or 'frequency'.
permute	Logical for permute columns in order to show seriation.
insert	Vector with the position after which you want to insert one or more hiatuses. Could be a list with two vectors: position and label to be printed instead of hiatus (see last examples).

Details

Seriograph

We have chosen the serigraph (DESACHY 2004). This tool makes it possible to highlight the evolution of ceramics over time as well as to understand the commercial relations thanks to the imported ceramics. The percentages of each category of ceramics per set are displayed. The percentages are calculated independently for each set (row). The display of the percentages allows comparison of the different sets but does not provide information on the differences in numbers. To fill this gap, the proportion of the numbers in each class is displayed on the seriograph (weight column).

We can generalize 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.

In order to facilitate the exploitation of the data tables, we propose here a computerised graphic processing tool (EPPM serigraph - for Ecart Positif aux Pourcentages Moyens - positive deviation from the average percentage), which does not require specialised statistical skills and is adapted to

the case of stratified sites, where the study of the evolution of artefacts can be based on the relative chronology provided by the excavation.

The treatment consists firstly of transforming this table of counts into a table of percentages, the total number in each set (each row) being reduced to 100; these are the proportions, or frequencies, of the types in the sets are thus compared.

The display of positive deviations from the mean percentages (EPPM) shows in black on a lighter background the percentage shares that are higher than the mean percentage of the variable, so as to highlight the most significant part of the values in the table. This display is simply adapted to the seriograph: when a percentage is greater than the average percentage of the type, the excess share (called here EPPM: positive deviation from the average percentage) is shown in black, centred around the axis of the type, on the grey background of the percentage bar.

The table is then transformed into a graphic matrix where these percentages are expressed, for each type, by horizontal bars centred on the column axis. When the rows are ordered chronologically, the silhouette formed by the superposition of these frequency bars makes it possible to visualise the evolution over time of the type concerned.

The display of the percentages allows comparison of the different sets but does not provide information on the differences in numbers. To fill this gap, the proportion of the numbers in each class is displayed on the seriograph (weight column).

The processing technique applies to sets whose chronological order is not known; the lines of the graph are to be reorganised so as to obtain boat-shaped silhouettes following the hypothesis of a chronological evolution corresponding to the seriation model.

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 (reorder)
frequency	Data frame of the frequencies data
EPPM	Data frame of EPPM (Positive Deviation from the Average Pourcentage)
EPM	Data frame of EPM (Deviation from the Average Pourcentage)

Author(s)

B. DESACHY
A. COULON
L. BELLANGER
P. HUSI

References

Desachy B. (2004). Le sériographe EPPM : un outil informatisé de sériation graphique pour tableaux de comptes. 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:10.3406/pica.2004.2396](https://doi.org/10.3406/pica.2004.2396)

Examples

```
data(datangkor)

## contingency table
cont <- datangkor$contingency

## default
seriograph(cont)

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

## Don't allow permutation of columns
seriograph(cont, permute = FALSE)

## insert Hiatus (position, 1 -> after first row from bottom: oldest)
```

```

seriograph(cont,insert = 2)
seriograph(cont,insert = c(2,3))

## insert costum label element
insert <- list(
  position = c(2,3),
  label = c("Hiatus.100years","Missing data")
)
seriograph(cont, insert = insert)

## change order with cluster name (letters on dendrogram) to sort them in a chronological order
seriograph(cont,order=c("AI03","AI09","AI01","AI02","AI04","APQR01","A005",
"A003","AI05","A001","APQR02","AI07","AI08","A002","AI06","A004","APQR03"))
## by omitting the row names, you delete the corresponding rows
seriograph(cont,order=c("AI09","AI06","AI08","AI05","AI07","AI04","AI03"))

## Compiègne data from Desachy (2004)
compiegne = data.frame(
A=c(13740, 13540, 12490, 6940, 6490),
B=c(375, 1520, 5255, 2880, 2350),
C=c(8270, 10110, 4220, 5800, 6900),
D=c(250, 740, 980, 3400, 2745),
E=c(20, 1230, 1395, 1510, 670),
F=c(10, 635, 1415, 2280, 3020),
G=c(40, 265, 440, 1080, 950),
H=c(80, 400, 680, 2840, 6700),
I=c(0, 0, 30, 620, 340),
J=c(5, 105, 580, 2075, 2660),
K=c(1740, 7210, 6750, 2130, 1080),
L=c(460, 1785, 5930, 2410, 570),
M=c(0, 410, 350, 910, 740),
N=c(1510, 565, 160, 410, 190),
O=c(350, 310, 10, 310, 985),
P=c(0, 450, 275, 410, 50)
)
rownames(compiegne) <- c("Période 5", "Période 4", "Période 3", "Période 2", "Période 1")

serio = seriograph(compiegne)
serio$contingency
serio$frequency
serio$EPPM
serio$EPM

```

serio_app

Launch the shiny application.

Description

see [seriograph](#). You can also check the wiki on the application.

Usage

```
serio_app()
```

Value

No return value

Author(s)

A. COULON
L. BELLANGER
P. HUSI

Examples

```
if(interactive()){
  serio_app()
}
```

timerange

Plot the time range of observations sorted by cluster.

Description

Cluster time range visualisation.

Usage

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

Arguments

data	data.frame containing the identifier, lower and upper bound (id, inf, sup) for each observation
cluster	vector containing the cluster number of each observation.
add	data.frame containing the information to be displayed on hover.
density	vector of the density for each observation.

color	vector of the colors for each observation (if you want the colors to correspond to something else than clusters). Character vector of the same length as the number of observations.
reorder	Logical to rearrange the colors. If TRUE, the first color corresponds to the leftmost cluster on the plot. If FALSE, the first color is that of cluster number 1, wherever it is.

Value

The function returns a list.

plot	The timerange plot.
------	---------------------

Author(s)

A. COULON
 B. MARTINEAU
 L. BELLANGER
 P. HUSI

Examples

```
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(1.1,1.1,1.1,1.1,1.1,1.1,1.2,1.2,1.2,1.2,1.2,2,2,2,2,2,2,2,2,2)

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

* datasets

datacerardat, 17

adjacency, 3

arrangePlot, 4, 10

arrondi, 6

bptest, 11, 13

CA, 9

CAdist, 8, 36

cerardat, 4, 9, 12, 13, 19, 39

cerardat_app, 12

cerardat_estim_nf, 9, 13

cutree, 24

datacancer, 15

dataceram, 16

datacerardat, 17

datangkor, 17

datarcheo, 18

dwtest, 11, 13

extract_results, 19

fastcluster::hclust, 20–22

hclust, 20, 37

hclustcompro, 22, 23, 41

hclustcompro_app, 23

hclustcompro_cutree, 22, 24

hclustcompro_select_alpha, 21–23, 25, 37

mapclust, 28, 31, 32

mapclust_app, 31

mapclust_cut_tree, 28, 32

optim_select_alpha, 21, 34

overlap, 35

par, 5

perioclust, 36, 38

perioclust_subdivide, 37, 38

plot.cerardat_obj, 10, 39

predict.lm, 10

rect.hclust, 24

rgb, 4, 19, 39

serio_app, 44

seriograph, 23, 41, 44

shapiro.test, 11, 13

silhouette, 28

SPARTAAS (SPARTAAS-package), 2

SPARTAAS-package, 2

stats::hclust, 20–22

text, 5, 39

timerange, 45

title, 4, 19, 24, 39

train, 13