Package: EDMAinR (via r-universe)

October 4, 2024

Type Package

Title Euclidean Distance Matrix Analysis in R

Version 0.3-0

Date 2023-08-21

Description A coordinate-free approach for comparing biological shapes using landmark data based on Lele and Richtsmeier (1991) <doi:10.1002/ajpa.1330860307>.

License GPL-2

LazyLoad yes

LazyData true

Imports pbapply, readxl, ape, shapes

Suggests knitr, rmarkdown, rgl, ggplot2, ggdendro, plotly

VignetteBuilder knitr

RoxygenNote 7.2.3

Encoding UTF-8

Repository https://psolymos.r-universe.dev

RemoteUrl https://github.com/psolymos/EDMAinR

RemoteRef HEAD

RemoteSha 4815a579113e33b75293adb85591fc24f116650e

Contents

EDMAinR-package *Euclidean Distance Matrix Analysis in R*

Description

A coordinate-free approach for comparing biological shapes using landmark data based on Lele and Richtsmeier (1991) <doi:10.1002/ajpa.1330860307>.

Details

EDMA data: `read_xyz`

Nonparametric fit: `edma_fit`

Form difference: `edma_fdm`

Growth and growth difference: `edma_gm`, `edma_gdm`

Shape difference: `edma_sdm`

Author(s)

Peter Solymos [aut, cre] (<https://orcid.org/0000-0001-7337-1740>), Subhash R. Lele [aut], Theodore M. Cole [aut], Liangyuan Hu [aut], Joan T. Richtsmeier [aut], Kevin M. Middleton [ctb] (<https://orcid.org/0000-0003-4704-1064>)

Maintainer: Peter Solymos <psolymos@gmail.com>

References

Lele, S. R., 1991. Some comments on coordinate-free and scale-invariant methods in morphometrics. American Journal of Physical Anthropology 85:407–417. <doi:10.1002/ajpa.1330850405>

Lele, S. R., and Richtsmeier, J. T., 1991. Euclidean distance matrix analysis: A coordinate-free approach for comparing biological shapes using landmark data. American Journal of Physical Anthropology 86(3):415–27. <doi:10.1002/ajpa.1330860307>

Lele, S. R., and Richtsmeier, J. T., 1992. On comparing biological shapes: detection of influential landmarks. American Journal of Physical Anthropology 87:49–65. <doi:10.1002/ajpa.1330870106>

Lele, S. R., and Richtsmeier, J. T., 1995. Euclidean distance matrix analysis: confidence intervals for form and growth differences. American Journal of Physical Anthropology 98:73–86. <doi:10.1002/ajpa.1330980107>

Hu, L., 2007. Euclidean Distance Matrix Analysis of Landmarks Data: Estimation of Variance. Thesis, Master of Science in Statistics, Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Alberta, Canada. Pp. 49.

Lele, S. R., and Cole, T. M. III., 1996. A new test for shape differences when variance-covariance matrices are unequal. Journal of Human Evolution 31:193–212. <doi:10.1006/jhev.1996.0057>

edma_colors *Check and manipulate colors*

Description

Check and manipulate the default color values.

Usage

```
edma_colors(n,
    type=c("diverging", "sequential", "qualitative"),
    alpha=1, rev=FALSE)

plot_edma_colors(n=9, maxq=9)
```

Arguments

n	the number of colors (>0) to be in the palette.
type	the type of palette.
alpha	the alpha transparency, a number in [0,1].
rev	logical, should colors be reversed.
maxq	maximum number of qualitative colors to plot.

Details

`edma_colors` create a vector of `n` colors based on the settings in `getOption("edma_options")`.

The options can be set via `options` (see Examples). The options can either be the name of a palette `hcl.colors`. When the option is set to multiple values, those are treated as colors to be interplolated with `colorRampPalette`. For qualitative palettes, the color values are used directly (recycled as needed).

Sequential palettes are produced as the higher half of the diverging palette for consistency.

Value

`edma_colors` returns a vector of hax color codes, `plot_edma_colors` produces a plot with the diverging, sequential, and qualitative default palettes given settings in `getOption("edma_options")`.

Author(s)

Peter Solymos

See Also

`hcl.colors`, `colorRampPalette`, `col2rgb`

Examples

```
## default palettes
plot_edma_colors(101)

## change default palettes
op <- options("edma_options" = list(
    diverging = "Green-Orange",
    qualitative = "Dark 2"))
plot_edma_colors(101)

## use color names
options("edma_options" = list(
    diverging = c("black", "grey", "pink"),
    qualitative = "Warm"))
plot_edma_colors(101)

## reset defaults
options(op)
plot_edma_colors(101)
```

---

edma_data *Functions for EDMA data objects*

Description

Functions for reading, simulating, and manipulating EDMA data.

Usage

```
## read xyz files
read_xyz(file, ...)

## write xyz files
write_xyz(x, file)

## data generation
edma_simulate_data(n, M, SigmaK)

## print
## S3 method for class 'edma_data'
print(x, truncate=40, ...)

## accessors
## S3 method for class 'edma_data'
dim(x)
## S3 method for class 'edma_data'
dimnames(x)
landmarks(x, ...)
```

```
dimensions(x, ...)
specimens(x, ...)
## S3 method for class 'edma_data'
landmarks(x, ...)
## S3 method for class 'edma_data'
dimensions(x, ...)
## S3 method for class 'edma_data'
specimens(x, ...)
landmarks(x) <- value
dimensions(x) <- value
specimens(x) <- value

## subsetting
## S3 method for class 'edma_data'
subset(x, subset, ...)
## S3 method for class 'edma_data'
x[i, j, k]

## coercion
## S3 method for class 'edma_data'
stack(x, ...)
## S3 method for class 'edma_data'
as.matrix(x, ...)
## S3 method for class 'edma_data'
as.data.frame(x, ...)
## S3 method for class 'edma_data'
as.array(x, ...)
as.edma_data(x, ...)
## S3 method for class 'array'
as.edma_data(x, ...)

combine_data(a, b,
    ga="G1", gb="G2")
combine_data4(a1, a2, b1, b2,
    ga1="A1", ga2="A2", gb1="B1", gb2="B2")

## plot methods
plot_2d(x, ...)
plot_ord(x, ...)
plot_clust(x, ...)
## S3 method for class 'edma_data'
plot(x, which=NULL,
    ask=dev.interactive(), ...)
## S3 method for class 'edma_data'
plot_2d(x, which=NULL, ...)
## S3 method for class 'edma_data'
plot_ord(x, ...)
## S3 method for class 'edma_data'
```

```
plot_clust(x, ...)

## dissimilarities
## S3 method for class 'edma_data'
as.dist(m, diag=FALSE, upper=FALSE)
```

Arguments

file	the name of the file which the data are to be read from, or written to, see read.table for more details.
x, m	an EDMA data object of class 'edma_data'.
which	if a subset of the specimens is required.
value	a possible value for dimnames(x).
ask	logical, if TRUE, the user is asked before each plot.
subset, i, j, k	subset is for subsetting specimens (e.g. for bootstrap). [i, j, k] indices refer to [landmarks, dimensions, specimens].
n, M, SigmaK	number of specimens (n), mean form matrix (M, K x D), variance-covariance matrix (K x K symmetric).
truncate	numeric, number of characters to print for the object title.
diag, upper	logical, indicating whether the diagonal and the upper triangle of the distance matrix should be printed. See as.dist.
a, b, a1, a2, b1, b2	EDMA data objects to be combined together. Landmarks must be homologous (determined by dimension names).
ga, gb, ga1, ga2, gb1, gb2	character, group names that are prepended to the specimen names to differentiate the groups.
...	other arguments passed to methods. For read_xyz, arguments passed to read.table.

Details

The xyz landmark data has the following structure, see Examples:

- Header: this is the description of the data.

- XYZ: indicates dimensions, XYZ means 3D landmark data.

- 42L 3 9: dimensions, e.g. 42 landmarks (K), 3 dimensions (D), 9 specimens (n).

- Landmark names, separated by space.

- The stacked data of landmark coordinates, e.g. 3 columns, space separated numeric values with K*n rows, the K landmarks per individuals stacked n times.

- Blank line.

- Date on of scans for each specimen (n rows), this part is also used to get specimen IDs.

After reading in or simulating and EDMA data object, the methods help extracting info, or manipulate these objects. See Values and Examples.

The EDMA data object (class 'edma_data') is a list with two elements: `$name` is the data set name (header information from the .xyz file), `$data` is a list of n matrices (the list can be named if speciemen information is present), each matrix is of dimension K x D, dimension names for the matrices describing landmark names and coordinate names.

Value

`edma_simulate_data` returns an EDMA data object of class 'edma_data'.

The `dim` returns the number of landmarks (K), dimensions (D), and specimens (n) in a data object.

`landmarks`, `dimensions`, and `specimens` are dimensions names, `dimnames` returns these as a list. Landmark names and dimensions are used to check if landmarks are homogeneous among objects. It is possible to set the dimansion names as `dimnames(x) <- value` where `value` is the new value for the name.

The `print` method prints info about the data object.

The methods `stack` and `as.matrix` return a stacked 2D array (K*n x D) with the landmark coordinates, `as.data.frame` turns the same 2D stacked array into a data frame, `as.array` returns a 3D array (K x D x n). `as.edma_data` turns a 3D array to an EDMA data object, this is useful to handle 3D array objects returned by many functions of the geomorph package (i.e. after reding Morphologika, NTS, TPS files).

`combine_data` and `combine_data4` combines 2 or 4 EDMA data sets together, landmarks must be homologous.

`as.dist` calculates the dissimilarity matrix (n x n, object of class 'dist', see `dist`) containing pairwise dissimilarities among the specimens. Dissimilarity is based on the T-statistic (max/min distance) averaged (so that it is symmetric) and on the log scale (so that self dissimilarity is 0).

`subset` and `[i,j,k]` returns an EDMA data object with the desired dimensions or permutations. See Examples.

`plot` and `plot_2d` produces a series of plots as a side effect, returning the data object invisibly. The functions provide diagnostics for each specimen or just the specimen selected by the `which` argument. The 2D projection is used in case of 3D landmark data. The convex hull of the specimens (excluding the one being selected) is compared with the actual specimen's landmarks. This allows easy spotting of erroneous data.

The `plot_ord` and `plot_clust` are based on the dissimilarities among specimens and provide ordination (metric multidimensional scaling using `cmdscale` based on square rooted dissimilarities and Cailliez's correction). and hierarchical cluster dendrogram (using the `hclust` function with Ward's clustering method).

Author(s)

Peter Solymos

See Also

`plot.edma_data` for visualizing EDMA data objects.

`edma_fit` for EDMA analysis.

`dist` for dissimilarity matrices and `global_test` for description of the T-statistic.

Examples

```
## read xyz files
file <- system.file(
    "extdata/crouzon/Crouzon_P0_Global_MUT.xyz",
    package="EDMAinR")
x <- read_xyz(file)
x

## test writing xyz file
f <- tempfile(fileext = ".xyz")
write_xyz(x, file=f)
tmp <- read_xyz(file=f)
stopifnot(identical(dimnames(x), dimnames(tmp)))
unlink(f)

## the orignal structure
l <- readLines(file)
cat(l[1:10], sep="\n")
cat(l[(length(l)-10):length(l)], sep="\n")

## plots
plot(x[,,1:5]) # steps through all individuals
plot_2d(x) # all speciemns in 1 plot
plot_2d(x, which=2) # show specimen #2
plot_ord(x)
plot_clust(x)

## dimensions and names
dim(x)
dimnames(x)
landmarks(x)
specimens(x)
dimensions(x)

## subsets
x[1:10, 2:3, 1:5]
subset(x, 1:10)

## coercion
str(as.matrix(x))
str(as.data.frame(x))
str(stack(x))
str(as.array(x))
as.edma_data(as.array(x))

## simulate data
K <- 3 # number of landmarks
D <- 2 # dimension, 2 or 3
sig <- 0.75
rho <- 0
SigmaK <- sig^2*diag(1, K, K) + sig^2*rho*(1-diag(1, K, K))
M <- matrix(c(0,1,0,0,0,1), 3, 2)
```

```
M[,1] <- M[,1] - mean(M[,1])
M[,2] <- M[,2] - mean(M[,2])
M <- 10*M

edma_simulate_data(10, M, SigmaK)
```

---

edma_fdm *Form difference*

Description

Form difference matrix based inference based on Lele and Richtsmeier (1992, 1995).

Usage

```
edma_fdm(numerator, denominator,
  B=0, ref_denom=TRUE, mix=FALSE)

get_influence(object, ...)
## S3 method for class 'edma_dm'
get_influence(object, level=0.95, ...)
## S3 method for class 'edma_influence'
plot(x, ...)

get_fdm(object, ...)
## S3 method for class 'edma_fdm'
get_fdm(object, sort=FALSE, level=0.95,
    what="all", ...)
global_test(object, ...)
## S3 method for class 'edma_fdm'
global_test(object, ...)
## S3 method for class 'edma_dm'
confint(object, parm, level=0.95, ...)

## S3 method for class 'edma_fdm'
print(x, ...)
## S3 method for class 'edma_fdm'
landmarks(x, ...)
## S3 method for class 'edma_fdm'
dimensions(x, ...)

plot_ci(x, ...)
plot_test(x, ...)
## S3 method for class 'edma_dm'
plot(x, ...)
## S3 method for class 'edma_dm'
plot_2d(x, ...)
```

```
## S3 method for class 'edma_dm'
plot_3d(x, ...)
## S3 method for class 'edma_dm'
plot_test(x, ...)
## S3 method for class 'edma_fdm'
plot_ci(x, ...)
## S3 method for class 'edma_fdm'
plot_ord(x, ...)
## S3 method for class 'edma_fdm'
plot_clust(x, ...)
```

Arguments

numerator, denominator	EDMA fit object to compare forms.
B	nonnegative integer, the number of bootstrap replicates.
ref_denom	logical, when TRUE, the denominator is used as reference object (its form matrix is fixed when calculating bootstrap comparing to the other object).
mix	logical, to use mixed bootstrap (numerator and denominator populations are mixed with replacement) or not (only the non-reference population is resampled with replacement, reference is fixed).
x, object	an EDMA FDM object of class 'edma_fdm'.
sort	logical, if stacked distances are to be sorted, see Examples.
level	numeric, between 0 and 1, alpha level for confidence interval.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. See confint.
what	what part of the ford differences to return: "all", "less" or "greater" than 1, "signif" or "nonsignif".
...	other arguments passed to methods.

Details

Form difference (FDM) is calculated as the ratio of form matrices (FM) from the numerator and denominator objects following Lele and Richtsmeier (1992, 1995): FDM(A,B) = FM(B)/FM(A). Form matrices are formed as pairwise Euclidean distances between landmarks from EDMA fit objects using the estimated mean forms.

Bootstrap distribution is based on either 'mixed' or not mixed bootstrap distribution. The 'mixed' bootstrap means that the bootstrap distribution represents n1+n2 specimens from the pooled sample of the numerator and denominator populations.

The default is mix=FALSE in which case we fix the reference FM and taking the ratio between the reference FM and the bootstrap FMs from the other non-reference object (depending on the ref_denom argument).

The T-statistic is based on the pairwise distanced in the FM, taking the max/min of the distances. Confidence intervals for local testing (via confint, get_fdm, and plot_ci) and T-test for global testing (via global_test, and plot_test) is based on the observed T-statistic and the bootstrap distribution.

The global testing algorithm is as follows: Suppose population 1 is the 'reference' population. Step 1: Resample n1 observations from the first sample and compute FM1*. Step 2: Resample n2 observations from the first sample and compute FM2*. Step 3: Compute the FDM* = FM2*/FM1* and T* = max(FDM*)/min(FDM*) Step 4: Repeat the above three steps B times to get the p-value.

Local testing (CI: confidence interval calculation) for elements of the FDM is based on the following algorithm: Step 1: Resample n1 observations from the first sample and compute FM1*. Step 2: Resample n2 observations from the second sample and compute FM2*. Step 3: Compute the FDM* = FM2*/FM1* Step 4: Repeat the above three steps B times to get the confidence intervals for the elements of the FDM.

Influential landmarks are identified by leaving one landmark out, then comparing the T-statistic with the value based on all the landmarks. The existing bootstrap distribution of the mean form is used (i.e. no re-estimation of the mean form) in `get_influence`.

Value

`edma_fdm` compares two EDMA fit objects and calculates form difference.

`confint` returns the confidence intervals for FDM, the `get_fdm` extract the stacked FDM with confidence intervals, the `plot_ci` visualizes the ordered form differences with confidence intervals.

`get_influence` extracts landmark influence information, the plot method visualizes this.

`global_test` presents the global T-test, the bootstrap distribution and observed T-value is visualized by `plot_test`.

`plot` and `plot_2d` produces a 2D plot of the mean form from the reference object ('prototype'). `plot_3d` use the rgl package to make a 3D plot using the same mean form. Influential landmarks are colored red. Lines represent distances between landmarks, <1 differences are colored blue, >1 differences are colored red.

The `plot_ord` and `plot_clust` produce plots based on dissimilarities among specimens in the two objects.

Author(s)

Peter Solymos, Subhash R. Lele, Theodore M. Cole, Joan T. Richtsmeier

References

Lele, S. R., and Richtsmeier, J. T., 1992. On comparing biological shapes: detection of influential landmarks. American Journal of Physical Anthropology 87:49–65. <doi:10.1002/ajpa.1330870106>

Lele, S. R., and Richtsmeier, J. T., 1995. Euclidean distance matrix analysis: confidence intervals for form and growth differences. American Journal of Physical Anthropology 98:73–86. <doi:10.1002/ajpa.1330980107>

See Also

Nonparametric fit: `edma_fit`

Growth difference: `edma_gdm`

Shape difference: `edma_sdm`

Examples

```
file1 <- system.file("extdata/crouzon/Crouzon_P0_Global_MUT.xyz",
    package="EDMAinR")
x1 <- read_xyz(file1)

file2 <- system.file("extdata/crouzon/Crouzon_P0_Global_NON-MUT.xyz",
    package="EDMAinR")
x2 <- read_xyz(file2)

numerator <- edma_fit(x1, B=10)
denominator <- edma_fit(x2, B=10)

fdm <- edma_fdm(numerator, denominator, B=10)
fdm2 <- edma_fdm(numerator, denominator, B=10, ref_denom=FALSE)
fdm
fdm2

head(get_fdm(fdm))
head(get_fdm(fdm, sort=TRUE, decreasing=TRUE))
head(get_fdm(fdm, sort=TRUE, decreasing=FALSE))

global_test(fdm)
global_test(fdm2)

head(confint(fdm))

head(infl <- get_influence(fdm))
plot(infl)

plot_ord(fdm)
plot_clust(fdm)
plot_test(fdm)
plot_ci(fdm)
plot_2d(fdm)
if (interactive())
    plot_3d(fdm)
```

`edma_fit` *Nonparametric EDMA fit*

Description

Estimate mean form and SigmaKstar matrix based on Lele (1991), Lele and Richtsmeier (1991) and Hu (2007).

Usage

```
edma_fit(x, B=0, ncores=getOption("Ncpus", 1L))
```

```
## generics
Meanform(object, ...)
SigmaKstar(object, ...)
get_fm(object, ...)

## methods
## S3 method for class 'edma_fit_np'
print(x, truncate=40, ...)
## S3 method for class 'edma_fit'
Meanform(object, ...)
## S3 method for class 'edma_fit'
SigmaKstar(object, ...)
## S3 method for class 'edma_fit'
get_fm(object, sort=FALSE, level=0.95, ...)
## S3 method for class 'edma_fit'
confint(object, parm, level=0.95, ...)
## S3 method for class 'edma_fit'
as.edma_data(x, ...)

## plot methods
plot_3d(x, ...)
## S3 method for class 'edma_fit'
plot(x, ...)
## S3 method for class 'edma_fit'
plot_2d(x, ...)
## S3 method for class 'edma_fit'
plot_3d(x, ...)
## S3 method for class 'edma_fit'
plot_ord(x, ...)
## S3 method for class 'edma_fit'
plot_clust(x, ...)

## distance manipulation
## S3 method for class 'edma_fit'
as.dist(m, diag=FALSE, upper=FALSE)
## S3 method for class 'dist'
stack(x, ...)
```

Arguments

x, object, m	an EDMA data object of class 'edma_data'.
B	nonnegative integer, the number of bootstrap replicates.
ncores	positive integer, the number of cores to use when bootstrapping. Use options(Ncpus = 2) to set it to 2 globally.
truncate	numeric, number of characters to print for the object title.
sort	logical, if stacked distances are to be sorted, see Examples.
level	numeric, between 0 and 1, alpha level for confidence interval.

parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. See confint.
diag, upper	logical, indicating whether the diagonal and the upper triangle of the distance matrix should be printed. See as.dist.
...	other arguments passed to methods. E.g. for plot_clust, the method describing the clustering agglomeration method to be used by the link{hclust} function (default is "ward.D2").

Details

The function estimates mean form and SigmaKstar matrix based on Lele (1991), Lele and Richtsmeier (1991) and Hu (2007).

Value

edma_fit returns and EDMA fit object of class 'edma_fit'. .edma_fit_np is the workhorse function behind edma_fit.

stack.dist takes any distance matrix of class 'dist' and turns that into a long form data frame with columns row and col indicating the row and column labels, dist giving the value in that cell. Only returns the values from the lower triangle of the matrix.

get_fm is the intended user interface to extract the form matrix (FM) from EDMA fit objects. This has the stacked distances based on the mean form. When the object has bootstrap replicates, get_fm also returns confidence intervals for the distances based on bootstrap and the confint method.

Meanform extracts the mean form (K x D) matrix, SigmaKstar extracts the corresponding uncertainties (K x K) based on the EDMA fit object.

plot and plot_2d produces a 2D plot of the mean form. 2D projection is used in case of 3D landmark data based on metric multidimensional scaling. plot_3d use the rgl package to make a 3D plot. The sizes of the dots correspond to square root of the SigmaKstar diagonal elements.

The plot_ord and plot_clust produce plots based on dissimilarities among specimens, see plot_ord.edma_data for details.

Author(s)

Peter Solymos, Subhash R. Lele, Theodore M. Cole, Liangyuan Hu, Joan T. Richtsmeier

References

Lele, S. R., 1991. Some comments on coordinate-free and scale-invariant methods in morphometrics. American Journal of Physical Anthropology 85:407–417. <doi:10.1002/ajpa.1330850405>

Lele, S. R., and Richtsmeier, J. T., 1991. Euclidean distance matrix analysis: A coordinate-free approach for comparing biological shapes using landmark data. American Journal of Physical Anthropology 86(3):415–27. <doi:10.1002/ajpa.1330860307>

Hu, L., 2007. Euclidean Distance Matrix Analysis of Landmarks Data: Estimation of Variance. Thesis, Master of Science in Statistics, Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Alberta, Canada. Pp. 49.

See Also

EDMA data: read_xyz

Form difference: edma_fdm

Growth difference: edma_gdm

Shape difference: edma_sdm

Examples

```
file <- system.file(
    "extdata/crouzon/Crouzon_P0_Global_MUT.xyz",
    package="EDMAinR")
x <- read_xyz(file)
x <- x[,,1:10] # 10 specimens

## nonparametric fit
fit <- edma_fit(x, B=9)
fit
str(Meanform(fit))
str(SigmaKstar(fit))

## form matrix
str(as.dist(fit))
str(stack(as.dist(fit)))

head(get_fm(fit))
head(get_fm(fit, sort=TRUE, decreasing=TRUE))
head(get_fm(fit, sort=TRUE, decreasing=FALSE))

plot_ord(fit)
plot_clust(fit)
plot(fit)
plot_2d(fit)
if (interactive())
    plot_3d(fit)
```

---

edma_gdm *Growth difference*

Description

Growth matrix and growth difference matrix based inference based on Lele and Richtsmeier (1992, 1995).

Usage

```
edma_gm(a1, a2, ...)
get_gm(object, ...)
## S3 method for class 'edma_gm'
```

```
get_gm(object, sort=FALSE, level=0.95,
    what="all", ...)

edma_gdm(a1, a2, b1, b2, ...)
get_gdm(object, ...)
## S3 method for class 'edma_gdm'
get_gdm(object, sort=FALSE, level=0.95,
    what="all", ...)

## S3 method for class 'edma_gm'
print(x, ...)
## S3 method for class 'edma_gdm'
print(x, ...)
## S3 method for class 'edma_gm'
global_test(object, ...)
## S3 method for class 'edma_gdm'
global_test(object, ...)
## S3 method for class 'edma_gdm'
landmarks(x, ...)
## S3 method for class 'edma_gdm'
dimensions(x, ...)

## S3 method for class 'edma_gdm'
plot_ord(x, ...)
## S3 method for class 'edma_gdm'
plot_clust(x, ...)
```

Arguments

a1, a2, b1, b2	EDMA fit object to compare growths.
x, object	an EDMA GM or GDM objects.
sort	logical, if stacked distances are to be sorted, see Examples.
level	numeric, between 0 and 1, alpha level for confidence interval.
what	what part of the ford differences to return: "all", "less" or "greater" than 1, "signif" or "nonsignif".
...	other arguments passed to edma_fdm, like ref_denom.

Details

Growth matrix (GM) is calculated as the ratio of form matrices (FM) from the numerator and denominator objects following Lele and Richtsmeier (1992, 1995): GM(A1,A2) = FM(A2)/FM(A1). Form matrices are formed as pairwise Euclidean distances between landmarks from EDMA fit objects using the estimated mean forms.

Growth difference matrix (GDM) is calculated as GDM(A1,A2,B1,B2) = GM(B1,B2) / GM(A1,A2).

Inference and visualization is similar to how it is done for FDMs.

Value

`edma_gm` compares two EDMA fit objects and calculates GM.

`edma_gdm` compares 4 EDMA fit objects and calculates GDM.

The `plot_ord` and `plot_clust` produce plots based on dissimilarities among specimens in the 2 or 4 objects (for GM and GDM, respectively).

Author(s)

Peter Solymos, Subhash R. Lele, Theodore M. Cole, Joan T. Richtsmeier

References

Lele, S. R., and Richtsmeier, J. T., 1992. On comparing biological shapes: detection of influential landmarks. American Journal of Physical Anthropology 87:49–65. <doi:10.1002/ajpa.1330870106>

Lele, S. R., and Richtsmeier, J. T., 1995. Euclidean distance matrix analysis: confidence intervals for form and growth differences. American Journal of Physical Anthropology 98:73–86. <doi:10.1002/ajpa.1330980107>

See Also

Nonparametric fit: `edma_fit`

Form difference: `edma_fdm`

Shape difference: `edma_sdm`

Examples

```
file_a1 <- system.file("extdata/growth/CZEM_wt_global.xyz",
    package="EDMAinR")
file_a2 <- system.file("extdata/growth/CZP0_wt_global.xyz",
    package="EDMAinR")

l <- c("amsph", "bas", "loci", "lpto", "lsqu",
        "lsyn", "roci", "rpto", "rsqu", "rsyn")

a1 <- read_xyz(file_a1)[l,,]
a2 <- read_xyz(file_a2)[l,,]
a1
a2

fit_a1 <- edma_fit(a1, B=10)
fit_a2 <- edma_fit(a2, B=10)

## --- growth matrix ---

gm <- edma_gm(a1=fit_a1, a2=fit_a2, B=10)
gm
global_test(gm)
head(confint(gm))
head(get_gm(gm))
```

```
head(get_gm(gm, sort=TRUE, decreasing=TRUE))
head(get_gm(gm, sort=TRUE, decreasing=FALSE))

plot_ord(gm)
plot_clust(gm)
plot_test(gm)
plot_ci(gm)
plot_2d(gm)
if (interactive())
    plot_3d(gm)

## --- growth difference matrix ---

file_b1 <- system.file("extdata/growth/CZEM_mut_global.xyz",
    package="EDMAinR")
file_b2 <- system.file("extdata/growth/CZP0_mut_global.xyz",
    package="EDMAinR")

b1 <- read_xyz(file_b1)[l,,]
b2 <- read_xyz(file_b2)[l,,]
b1
b2

fit_b1 <- edma_fit(b1, B=10)
fit_b2 <- edma_fit(b2, B=10)

gdm <- edma_gdm(a1=fit_a1, a2=fit_a2, b1=fit_b1, b2=fit_b2, B=10)
gdm
global_test(gdm)
head(confint(gdm))
head(get_gdm(gdm))
head(get_gdm(gdm, sort=TRUE, decreasing=TRUE))
head(get_gdm(gdm, sort=TRUE, decreasing=FALSE))

plot_ord(gdm)
plot_clust(gdm)
plot_test(gdm)
plot_ci(gdm)
plot_2d(gdm) # need real data
if (interactive())
    plot_3d(gdm)
```

`edma_scale` *Scale an EDMA data object*

Description

This function implements the landmark scaling procedures described in Lele and Cole (1996) which are used to rescale the landmarks for specimens in which the variance-covariance matrices to two populations are unequal.

Usage

```
edma_scale(x, scale_by, L1 = NULL, L2 = NULL, scale_constant = NULL)
```

Arguments

x	an EDMA data object of class edma_data.
scale_by	string specifying the type of scaling. Valid options are "constant", "endpoints", "geometric_mean", "maximum", "median", "sneath". See below for details.
L1	string specifying first landmark to use if scale_by = "endpoints"
L2	string specifying second landmark to use if scale_by = "endpoints"
scale_constant	numeric specifying the scaling constant to use for scale_by = "constant"

Details

scale_by determines the interlandmark scaling value. Options are:

- constant Interlandmark distances are scaled by a numeric constant that is applied to all specimens.
- endpoints Interlandmark distances are scaled by the distance between a pair of landmarks (L1 and L2) for each specimen.
- geometric mean Interlandmark distances are scaled by th geometric mean of all pairwise distances for each specimen.
- maximum Interlandmark distances are scaled by the maximum of all pairwise distances for each specimen.
- median Interlandmark distances are scaled by the median of all pairwise distances for each specimen.
- sneath Interlandmark distances are scaled using the method described by Sneath (1967), which uses the square-root of the mean squared distances of each landmark to the centroid. Also see Creel (1986).

Value

object of class 'edma_data', with landmarks scaled according to scale_by parameter. See details for details of scaling procedures. The object x are appended with a list including the the scaling method string and the values used for scaling. The latter can be useful for comparisons, e.g., of geometric means.

References

Creel, N. 1986. Size and Phylogeny in Hominoid Primates. *Syst. Zool.* 35:81-99.

Lele, S., and T. M. Cole III. 1996. A new test for shape differences when variance-covariance matrices are unequal. *J. Hum. Evol.* 31:193-212.

Sneath, P. H. A. 1967. Trend-surface analysis of transformation grids. *J. Zool.* 151:65-122. Wiley.

Examples

```
# Following the example in Lele and Cole (1996)
X <- matrix(c(0, 0, 2, 3, 4, 1), byrow = TRUE, ncol = 2)
Y <- matrix(c(0, 0, 3, 3, 3, 0), byrow = TRUE, ncol = 2)

# Bind matrices into 3d array and convert to edma_data
XY <- as.edma_data(array(dim = c(3, 2, 2),
                         data = cbind(X, Y)))

# Scale by a constant
XY_const <- edma_scale(XY, scale_by = "constant", scale_constant = 2)
print(XY_const)
XY_const$data
XY_const$scale

# Scale by distance between two landmarks
XY_endpt <- edma_scale(XY, scale_by = "endpoints", L1 = "L1", L2 = "L3")
print(XY_endpt)
XY_endpt$data
XY_endpt$scale

# Scale by geometric mean of all interlandmark distances
XY_geomean <- edma_scale(XY, scale_by = "geometric_mean")
print(XY_geomean)
XY_geomean$data
XY_geomean$scale

# Scale by maximum of all interlandmark distances
XY_max <- edma_scale(XY, scale_by = "maximum")
print(XY_max)
XY_max$data
XY_max$scale

# Scale by median of all interlandmark distances
XY_median <- edma_scale(XY, scale_by = "median")
print(XY_median)
XY_median$data
XY_median$scale

# Scale using root mean squared distance from each landmark to
# the centroid (Sneath, 1967).
XY_sneath <- edma_scale(XY, scale_by = "sneath")
print(XY_sneath)
XY_sneath$data
XY_sneath$scale
```

---

edma_sdm *Shape difference*

Description

Shape difference matrix based inference following Lele and Cole (1996).

Usage

```
edma_sdm(a, b, log=TRUE, size=TRUE, edge = NULL)
get_sdm(object, ...)
## S3 method for class 'edma_sdm'
get_sdm(object, sort=FALSE,
    level = 0.95, ...)

## S3 method for class 'edma_sdm'
print(x, level = 0.95, ...)
Z_test(object, ...)
## S3 method for class 'edma_sdm'
Z_test(object, level = 0.95, ...)
## S3 method for class 'edma_sdm'
landmarks(x, ...)
## S3 method for class 'edma_sdm'
dimensions(x, ...)

## S3 method for class 'edma_sdm'
confint(object, parm, level=0.95, ...)
## S3 method for class 'edma_sdm'
get_influence(object, statistic=c("Z", "C"),
    level=0.95, ...)

plot_Ztest(x, ...)
## S3 method for class 'edma_sdm'
plot_Ztest(x, statistic=c("Z", "C"),
    level = 0.95, ...)
```

Arguments

a, b	EDMA fit object to compare shapes.
x, object	a SDM object.
log	logical, if form matrix is to be log transformed before calculating the differences.
size	logical, if size difference (C) is to be estimated (TRUE) of fixed as 1 (FALSE).
edge	numeric or character, numeric IDs or the name of the 2 landmarks to be used to calculate C (C=db/da, where da and db are the edge distances between the two landmarks for object a and b respectively). C is calculated using total least squares (TLS) when edge=NULL.
sort	logical, if stacked distances are to be sorted, see Examples.
level	numeric, between 0 and 1, alpha level for confidence interval.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. See confint.

statistic	character, the Z or C statistic to be plotted.
...	other arguments passed to other functions.

Details

Shape difference matrix (SDM) is defined as the difference between the scaled form matrices S(A) and S(B). S(A) = C * FM(A), S(B) = FM(B), where C is a scaling factor and is calculated using total least squares (TLS). Shape difference matrix is S(A) - S(B) when log=FALSE and log(S(A)) - log(S(B)) when log=TRUE.

Inference and visualization is similar to how it is done for FDMs.

Note: the original implementation is usinga particular edge to calculate the size (C) parameter (size=TRUE and edge not NULL). edge=NULL uses total least squares to estimate C based on all the edges of all the landmarks. When size=FALSE we set C=1, assuming sizez are the same.

Value

edma_sdm compares 2 EDMA fit objects and calculates SDM.

Author(s)

Peter Solymos, Subhash R. Lele, Theodore M. Cole

References

Lele, S. R., and Cole, T. M. III., 1996. A new test for shape differences when variance-covariance matrices are unequal. Journal of Human Evolution 31:193–212. <doi:10.1006/jhev.1996.0057>

See Also

Nonparametric fit: edma_fit

Form difference: edma_fdm

Growth difference: edma_gdm

Examples

```
file_a <- system.file("extdata/growth/CZP0_wt_global.xyz",
    package="EDMAinR")
file_b <- system.file("extdata/growth/CZP0_mut_global.xyz",
    package="EDMAinR")
l <- c("amsph", "bas", "loci", "lpto", "lsqu",
        "lsyn", "roci", "rpto", "rsqu", "rsyn")

a <- read_xyz(file_a)[l,,]
b <- read_xyz(file_b)[l,,]
a
b

fit_a <- edma_fit(a, B=10)
fit_b <- edma_fit(b, B=10)
```

```
sdm <- edma_sdm(a=fit_a, b=fit_b)
sdm
Z_test(sdm)
head(confint(sdm))
head(get_sdm(sdm))
head(get_sdm(sdm, sort=TRUE, decreasing=TRUE))
head(get_sdm(sdm, sort=TRUE, decreasing=FALSE))

get_influence(sdm)

plot_Ztest(sdm, "Z")
plot_Ztest(sdm, "C")
plot_ci(sdm)

plot(get_influence(sdm))
```

---

gpa_fit *GPA*

Description

Fit GPA or WGPA to landmark data.

Usage

```
gpa_fit(x, B = 0, ncores = getOption("Ncpus", 1L),
    weighted=FALSE, ...)
## S3 method for class 'gpa_fit'
print(x, truncate=40, ...)
```

Arguments

x	an EDMA data object of class 'edma_data'.
B	nonnegative integer, the number of bootstrap replicates.
weighted	logical, use shapes::procWGPA instead of shapes::procGPA.
ncores	positive integer, the number of cores to use when bootstrapping. Use options(Ncpus = 2) to set it to 2 globally.
truncate	numeric, number of characters to print for the object title.
...	arguments passed to shapes::procGPA or shapes::procWGPA.

Value

Returns only form matrix, SigmaKstar is NA.

Author(s)

Peter Solymos wrote the wrapper for shapes::procGPA.

References

Gower, J.C. (1975). Generalized Procrustes analysis, Psychometrika, 40, 33–50.

Examples

```
file <- system.file(
    "extdata/crouzon/Crouzon_P0_Global_MUT.xyz",
    package="EDMAinR")
x <- read_xyz(file)
x <- x[,,1:10] # 10 specimens

## nonparametric fit
fit <- gpa_fit(x, B=9)
fit
str(Meanform(fit))
str(SigmaKstar(fit))
```

---

report *EDMA Report*

Description

Writes output into a text file following close the WinEDMA implementation.

Usage

```
edma_fdm_report(numerator, denominator, output="edma_output.txt",
  landmarks=NULL, B=0, level=0.95, ref_denom=TRUE, mix=FALSE,
  digits=4)

edma_gdm_report(numerator_yng, numerator_old,
    denominator_yng, denominator_old, output="edma_output.txt",
    landmarks=NULL, B=0, level=0.95, ref_denom=TRUE, mix=FALSE,
    digits=4)
```

Arguments

numerator, denominator, numerator_yng, numerator_old, denominator_yng, denominator_old	input file names or EDMA data objects.
output	path and file name for the output file.
landmarks	a subset of landmarks to be specified, or NULL (use all landmarks).
B	number of bootstrap samples.

level	confidence level.
ref_denom	logical, when TRUE, the denominator is used as reference object (its form matrix is fixed when calculating bootstrap comparing to the other object).
mix	logical, to use mixed bootstrap (numeriator and denominator populations are mixed with replacement) or not (only the non-reference population is resampled with replacement).
digits	significant digits to print.

Examples

```
## Not run:

edma_fdm_report(
    numerator = system.file(
        "extdata/crouzon/Crouzon_P0_Global_NON-MUT.xyz",
        package="EDMAinR"),
    denominator = system.file(
        "extdata/crouzon/Crouzon_P0_Global_MUT.xyz",
        package="EDMAinR"),
    output="edma_output.txt",
    landmarks=c("locc", "lpfl", "lpsq", "lpto", "lsqu",
        "rocc", "rpfl", "rpsq", "rpto", "rsqu"),
    B=1000,
    level=0.9,
    ref_denom=FALSE,
    mix=TRUE)


## End(Not run)
```

Index