

Package ‘lmkCHK’

October 1, 2020

Title Tools to Visualize and Screen 2D/3D Landmark Data

Version 1.0.0

Description Provides visual and interactive tools to screen 2D/3D landmark data.
Meant to supplement other geometric morphometric packages by making vibrant visuals easy/automatic.

Depends R (>= 3.2.0), geomorph, rgl

Imports Morpho, shiny

Suggests psych

License GPL (>= 3.0)

URL <<https://github.com/ehrichd/lmkCHK>>

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Daniel Ehrlich [aut, cre],
Michelle Singleton [aut]

Maintainer Daniel Ehrlich <dan.ehrlich.e@gmail.com>

Repository CRAN

Date/Publication 2020-10-01 09:40:07 UTC

R topics documented:

lmkCHK	2
LMK_bscale	2
LMK_colpick	3
LMK_colramp	4
LMK_compare_two	5
LMK_CVA_2D	5
LMK_CVA_3D	6
LMK_ellipse	6
LMK_euD	7

LMK_get_sliders 8

LMK_grp_apply 9

LMK_iccA 9

LMK_limset 10

LMK_obs_error 10

LMK_PCA_plot 11

LMK_plotoutliers 12

LMK_plotVar 12

LMK_Screen 13

LMK_swap 14

LMK_sym 14

LMK_viz 15

LMK_wireframe 16

LMK_writeland_nts 17

Index 18

lmkCHK

Landmark Check

Description

Helper functions for visualizing and screening 2D/3D landmark data.

Author(s)

Daniel E. Ehrlich

LMK_bscale

Backscale an array

Description

Scale each individual of a landmark array by an arbitrary value (e.g. centroid size)

Usage

LMK_bscale(A, cs)

Arguments

- A
- A 3D landmark array
- cs
- a vector of centroid sizes

Value

Returns a scaled 3D landmark array

Examples

```
require(geomorph)
data(plethodon)

plotAllSpecimens(plethodon$land) ## plot all unaligned specimens
gpa <- gpagen(plethodon$land) ## align AND SCALE data
plotAllSpecimens(gpa$coords)

backscaled <- LMK_bscale(gpa$coords, gpa$Csize)

plotAllSpecimens(backscaled) ## data are aligned, but in their original size
```

LMK_colpick

*Graphical Color Picker***Description**

Interactive function to choose precise color from a pallette of tints or shades

Usage

```
LMK_colpick(n = 1, type = c("shades", "tints"), transparency = 1)
```

Arguments

n	Number of colors to pick
type	One of c("shades", "tints") to plot
transparency	Set color transparency (alpha) from 0 (invisible) to 1 (opaque). Default is opaque

Details

Select any number of colors from palette

Value

Returns a matrix of n colors listed in 3 formats: Rcolor, RGB, and HSV.

LMK_colramp*Color Ramp*

Description

Create a color ramp based on a grouping factor

Usage

```
LMK_colramp(grp, mute = TRUE)
```

Arguments

<code>grp</code>	A grouping variable, should be a factor
<code>mute</code>	Logical, should color be muted. If true (default), color values will be generated with saturation value of .4, if false, values will have full saturation (1.0)

Details

Creates color ramp for a given factor and returns a vector of color assignments for each individual using `grDevices::rainbow()`. Useful for plotting by group.

Value

Returns a vector of color assignments for each individual based on a grouping factor.

Author(s)

Daniel Ehrlich

Examples

```
data <- cbind(1:20, 20:1)
grps <- as.factor(rep(c("A", "B", "C", "D", "E"), 4))
plot(data, col = LMK_colramp(grps), pch = 16) ##convert factor to rainbow pallete
```

LMK_compare_two	<i>Compare Two landmark arrays</i>
-----------------	------------------------------------

Description

Plot 2 landmark configurations in 3D space.

Usage

```
LMK_compare_two(A1, A2, new = FALSE, main = NULL)
```

Arguments

A1	A 3D array of landmark data; A reference
A2	A second 3D array of landmark data to compare to the reference
new	Should a new graphing window be opened (default is to overwrite the previous)
main	A character string to display as the title. If main= NULL (default) function will attempt to deparse object names.

Details

Presumably, this will be procrustes aligned coordinates, but any 2 p x k x n arrays will work

LMK_CVA_2D	<i>CVA 2D Shape Change</i>
------------	----------------------------

Description

Visualize Shape change in two dimensions of Morpho object.class 'CVA'

Usage

```
LMK_CVA_2D(CVA, variate = 1, scale = 5)
```

Arguments

CVA	object of class CVA generated by Morpho::CVA
variate	The canonical variate to be visualized (default = 1)
scale	A scaling factor to emphasize shape change (default = 5)

Value

Plots the mean landmark configuration of a CVA an vectors representing the shape change along a given variate, by a given scale

LMK_CVA_3D

*CVA 3D Shape Chance***Description**

Visualize Shape change in three dimensions of Morpho object.class 'CVA'

Usage

```
LMK_CVA_3D(CVA, variate = 1, scale = 5, links = NULL)
```

Arguments

CVA	object of class CVA generated by Morpho::CVA
variate	The canonical variate to be visualized (default = 1)
scale	A scaling factor to emphasize shape change (default = 5)
links	A wireframe for visualizing. WIP

Value

Plots the mean landmark configuration of a CVA an vectors representing the shape change along a given variate, by a given scale

LMK_ellipse

*Confidence Ellipses***Description**

Draw confidence ellipses around data

Usage

```
LMK_ellipse(
  dat,
  ci = c(67.5, 90, 95, 99),
  linesCol = "black",
  fillCol = "grey",
  smoothness = 20
)
```

Arguments

<code>dat</code>	data to be plotted in matrix form [X,Y]
<code>ci</code>	Confidence interval to be plotted. Must be one of c(67.5, 90, 95,99)
<code>linesCol</code>	Color for the line. Currently takes hsv() format
<code>fillCol</code>	color for the fill. Currently only takes hsv() format, set NULL for no fill
<code>smoothness</code>	Smoothness for ellipses. Default should be sufficient but is customizable.

Examples

```
set.seed(1)

datx <- rnorm(100,0,1)
daty <- rnorm(100,0,10)

dat <- cbind(datx, daty)

plot(dat)
LMK_ellipse(dat, ci = 90)
```

LMK_euD

*Euclidean Distance***Description**

Calculate Euclidean distance between two sets of 3D landmark data. Data should be in the typical $p \times k \times n$ array.

Usage

```
LMK_euD(A1, A2)
```

Arguments

<code>A1</code>	A 3D landmark array in the form $p \times k \times n$
<code>A2</code>	A second 3D landmark array in the form $p \times k \times n$

Value

Returns a list (3L) containing the direct Euclidean distance for all individuals/landmarks, as well as summarized by individual and landmark

Examples

```
lmks1 <- cbind(1:10, 1:10, 1:10)
lmks2 <- lmks1*10

LMK_euD(lmks1,lmks2) ## distance between each landmark (1 through 10)

LMK_euD(lmks1[1,], lmks2[1,]) ## distance between a single landmark
```

LMK_get_sliders	<i>Get sliders</i>
-----------------	--------------------

Description

Define a sliders table (geomorph) for a contour with fixed endpoints

Usage

```
LMK_get_sliders(c1)
```

Arguments

`c1` A vector of landmark numbers

Details

Assumes all points slide sequentially between endpoints

Value

Returns a table of sliders suitable for use with `geomorph::gpagen()`. This function assumes all points slide sequentially between the max/min landmarks in the sequence.

Author(s)

Daniel Ehrlich

Examples

```
lmknums <- 1:100

sli <- LMK_get_sliders(lmknums[80:100]) ## to treat landmarks 80 through 100 as sliders

sli
```

LMK_grp_apply

*Group Apply***Description**

Apply a function (f) to a set of data (d) by a grouping variable (grp). Calculates values within group (eg mean), as well as for the entire ungrouped sample (eg, grand mean).

Usage

```
LMK_grp_apply(d, f, grp)
```

Arguments

d	a dataset
f	a function
grp	a grouping variable

Examples

```
data(iris)
LMK_grp_apply(d = iris[,1:4], f = mean, grp = iris$species)
```

LMK_iccA

*Intra class correlations and Chronbach's Alpha***Description**

Compute ICC and CA for each dimension (X,Y,Z) of arrays

Usage

```
LMK_iccA(obs1, obs2)
```

Arguments

obs1	a 3D landmark array
obs2	a 3D landmark array

Value

Returns a table of ICC coefficients, associated p-values, and Chronbach's Alpha

LMK_limset	<i>limset</i>
------------	---------------

Description

Function to easily set plot buffer

Usage

```
LMK_limset(x, factor = 1.2)
```

Arguments

x	A vector to be plotted
factor	to expand the data (default=1.2)

Details

Takes a vector of data, calculates the range of data and scales it by a factor. Returns new range of length=2

Examples

```
vec <- 1:10
extend_vec <- LMK_limset(vec)
extend_vec
```

LMK_obs_error	<i>Inter-/Intra-Observer Error</i>
---------------	------------------------------------

Description

Calculate the inter- or intra-observer error between two sets of LMK data

Usage

```
LMK_obs_error(obs1, obs2, lmk.lbl = NULL, full = FALSE)
```

Arguments

obs1	A set of observations of landmark data
obs2	A second set of observations of landmark data
lmk.lbl	A character vector containing informative names for landmarks. If Null (default) LMKs will be labeled as 'lmk.1, lmk.2, lmk.3, ...'
full	Logical, should a full output be returned. By default (FALSE), only a summary table is returned. If TRUE, a list (3L) is returned with additional information.

Details

Computes three measures of error between two sets of LMK data. Direct Euclidean distance, intraclass correlation (ICC), and Chronbach's alpha. All values are summarized by landmark, and by individual to investigate/identify systemic error. Output is a summary table, or a list (3L) containing the summary table, as well as additional raw and transformed values.

Value

Test returns a table summarizing euclidean distance, ICC and Chronbach's alpha. If full = TRUE, then the function returns a list (3L) of additional raw/transformed values used in calculations

Author(s)

Daniel Ehrlich

LMK_PCA_plot

Shapespace PCA

Description

Plot 2D or 3D scatter plot of tangent shape-space for a GPA object (of gpagen())

Usage

```
LMK_PCA_plot(A, xPC = 1, yPC = 2, zPC = NULL, grp)
```

Arguments

A	A 3d landmark array
xPC	an integer indicating which PC to plot, default = 1
yPC	an integer indicating which PC to plot, default = 2
zPC	an integer indicating which PC to plot, default = NULL (2D)
grp	a grouping factor

LMK_plotoutliers	<i>Plot Outliers</i>
------------------	----------------------

Description

Function to identify outliers visually

Usage

```
LMK_plotoutliers(A, gpa = TRUE, plotALL = TRUE, ...)
```

Arguments

A	Raw 3D coordinate data, procrustes aligned coordinate data, or an object of class <code>gpgen</code>
gpa	Logical, should a GPA be calculated. If TRUE, A is assumed to be raw coordinate data and GPA is implemented via <code>geomorph::gpgen</code> . If FALSE, no imposition will be done. If TRUE, gpa-aligned coordinates and mshape will also be exported
plotALL	Logical, plot all specimens (and mean) in 3D space
...	Additional parameters to pass to <code>gpgen</code>

Value

Returns a list containing summary information of the dataset, individual procrustes distances, GPA aligned Procrustes Coordinates and mean shape configuration

Examples

```
require(geomorph)
data(scallops)

outliers <- LMK_plotoutliers(scallops$coorddata, plotALL = FALSE) ## skip 3D plot
outliers$summary.info
```

LMK_plotVar	<i>Plot Variance</i>
-------------	----------------------

Description

Function to plot variance within landmarks for a dataset

Usage

```
LMK_plotVar(a, lmknames = NULL, tabonly = FALSE, ...)
```

Arguments

<code>a</code>	an array, presumably already aligned
<code>lmknames</code>	A vector of landmark names to plot. If null (default) points are labeled by index.
<code>tabonly</code>	Logical. If FALSE (default) function returns list of mean variance by landmark and plots values. If TRUE, plotting steps are suppressed.
<code>...</code>	Additional arguments to pass to plotting functions

Value

Returns a list of mean variance by landmark for the entire dataset. Also produces a plot with mean variance (green) and 1 SD (dark green)

Examples

```
require(geomorph)
data(plethodon)
gpa <- gpagen(plethodon$land)
LMK_plotVar(gpa$coords) ## most landmarks show low variance
## lms 11,12 plot above 1 standard deviation
```

LMK_Screen

Launch interactive screening plots

Description

GUI interface to choose DTA file, and various classifiers to conduct preliminary analysis.

Usage

```
LMK_Screen()
```

LMK_swap

*Swap Landmarks***Description**

Function to swap landmark sequences within an individual or across an entire array

Usage

```
LMK_swap(a, l1, l2)
```

Arguments

a	a p x k x n matrix of landmark coordinates
l1	a numeric vector of landmark index (sequence number) to be changed
l2	a numeric vector of landmark index (Sequence number) to change to

Author(s)

Daniel Ehrlich

Examples

```
lmks <- cbind(1:10, 1:10, 1:10)
plot(lmks, col = rainbow(10), pch = 16) ## plot sequence (in 2 dimensions)
lmks2 <- LMK_swap(lmks, l1 = c(2,8), l2 = c(8,2)) ## flip landmarks 2,8
plot(lmks2, col = rainbow(10), pch = 16) ## show flipped landmarks
```

LMK_sym

*Symmetrize***Description**

Function to provide easy specification and handling of multistep symmetrization process

Usage

```
LMK_sym(A, Lmpair)
```

Arguments

A	A landmark configuration in the form p x k x n
Lmpair	A matrix listing paired landmarks, those not listed are assumed to be midline

Details

Under full sequence landmarks are first estimated using bilateral reflection. If missing data are still present, landmarks are estimated using TPS interpolation of n=3 nearest neighbors. Now complete LMK arrays are reflected a final time and averaged creating perfectly symmetrical configurations

Value

Returns a symmetrized landmark configuration with no missing data

LMK_viz

Shape Change Visualizations

Description

Easily visualize shape change and create relative warps for LMK data

Usage

```
LMK_viz(dat, ...)
```

Arguments

dat	Some form of landmark data of class array (p x k x n); gpagen (geomorph); PCA/CVA (morpho)
...	Additional arguments to pass to subsequent functions based on input data

Details

Attempt to combine all of these into a single function

Value

will return a list of helpful information and some nice plots

LMK_wireframe	<i>Compare Wireframes</i>
---------------	---------------------------

Description

Function to plot two landmark configurations as wireframes, and compare directly

Usage

```
LMK_wireframe(
  A1,
  A2,
  cols = c("red", "blue"),
  links = NULL,
  pts = c("point", "sphere", "none"),
  rad = c(0.001, 0.001),
  vectors = FALSE,
  ...
)
```

Arguments

A1	a reference
A2	a target
cols	a list of two colors to use for plots (default Red, Blue)
links	a wireframe listing landmarks to be linked
pts	Should landmarks be plotted as points, spheres, or not at all.
rad	radius to plot for spheres, default is .001. Unclear how project specific
vectors	Should vectors be drawn linking homologous landmarks. This may help show the direction of change, or could look clunky
...	Additional arguments to pass to plotting functions.

Details

Plots two LMK configurations on top of each other as some combination of fixed points and wire-frame. Presumably this will be PC min/max, two group means, or mshape and a target individual, but any two matching configurations should work

LMK_writeland_nts	<i>Write DTA/NTS</i>
-------------------	----------------------

Description

Write 3D array as a .dta/.nts file

Usage

```
LMK_writeland_nts(A, filepath)
```

Arguments

A	a 3D array in the form of [p x k x n]
filepath	Filename/Filepath, including extension: .dta, or .nts

Author(s)

Daniel Ehrlich

Index

LMK_bscale, [2](#)
LMK_colpick, [3](#)
LMK_colramp, [4](#)
LMK_compare_two, [5](#)
LMK_CVA_2D, [5](#)
LMK_CVA_3D, [6](#)
LMK_ellipse, [6](#)
LMK_euD, [7](#)
LMK_get_sliders, [8](#)
LMK_grp_apply, [9](#)
LMK_iccA, [9](#)
LMK_limset, [10](#)
LMK_obs_error, [10](#)
LMK_PCA_plot, [11](#)
LMK_plotoutliers, [12](#)
LMK_plotVar, [12](#)
LMK_Screen, [13](#)
LMK_swap, [14](#)
LMK_sym, [14](#)
LMK_viz, [15](#)
LMK_wireframe, [16](#)
LMK_writeland_nts, [17](#)
lmkCHK, [2](#)