Package: paleomorph (via r-universe)

September 4, 2024

Type Package

Title Geometric Morphometric Tools for Paleobiology

Version 0.1.4

Date 2017-04-19

Author Tim Lucas, Anjali Goswami

Maintainer Tim Lucas <timcdlucas@gmail.com>

Description Fill missing symmetrical data with mirroring, calculate Procrustes alignments with or without scaling, and compute standard or vector correlation and covariance matrices (congruence coefficients) of 3D landmarks. Tolerates missing data for all analyses.

License MIT + file LICENSE

Imports stats, utils

Suggests knitr, testthat, abind, rgl

LazyData true

URL https://github.com/timcdlucas/paleomorph/

BugReports https://github.com/timcdlucas/paleomorph/issues

RoxygenNote 5.0.1

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

RemoteUrl https://github.com/timcdlucas/paleomorph

RemoteRef HEAD

RemoteSha 681e75f367bc77e1fc454ab484a49945dda72905

Contents

`countMissing` *Count the number of missing landmarks in an array*

Description

Count the number of missing landmarks in an array

Usage

```
countMissing(A)
```

Arguments

`A`	An N x 3 x M array where N is the number of landmarks, 3 is the number of dimensions, and M is the number of specimens.

Value

A length n vector giving the number of missing landmarks for each specimen.

Examples

```
A <- array(1:(3*6*7), dim = c(7, 3, 6))
A[2, , 1] <- NA
countMissing(A)
```

`covar` *Calculate covariance matrix between individual landmark coordinates*

Description

Calculate covariance matrix between individual landmark coordinates. Skips any missing values in computation of covariance matrix.

Usage

```
covar(A)
```

Arguments

A	An N x 3 x M array where N is the number of landmarks, 3 is the number of dimensions, and M is the number of specimens.

Details

This function does not guarantee that the returned matrix is positive definite. If the covariance matrix is not positive definite a warning is given and the matrix can be bent to create the closest positive definite matrix with `as.matrix(Matrix::nearPD(mat)$mat)`.

Value

3N x 3N covariance matrix

Examples

```
A <- array(rnorm(4 * 2 * 3), dim = c(2, 3, 4))
A.cov <- covar(A)
```

---

dotcorr *Calculate 3D vector correlation matrix using the congruence coefficient. Skips any missing values in computation of correlation matrix*

Description

Calculate 3D vector correlation matrix using the congruence coefficient. Skips any missing values in computation of correlation matrix. Gives an N x N correlation matrix.

Usage

```
dotcorr(A)
```

Arguments

A	An N x 3 x M array where N is the number of landmarks, 3 is the number of dimensions, and M is the number of specimens.

Value

Correlation matrix

Examples

```
A <- array(rnorm(4 * 2 * 3), dim = c(2, 3, 4))
A.corr <- dotcorr(A)
```

`dotcvm` *Calculate 3D covariance matrix using unscaled congruence coefficient. Skips any missing values in computation of covariance matrix*

Description

Calculate 3D covariance matrix using unscaled congruence coefficient. Skips any missing values in computation of covariance matrix

Usage

```
dotcvm(A)
```

Arguments

`A`	An N x 3 x M array where N is the number of landmarks, 3 is the number of dimensions, and M is the number of specimens.

Details

This function does not guarantee that the returned matrix is positive definite. If the covariance matrix is not positive definite a warning is given and the matrix can be bent to create the closest positive definite matrix with `as.matrix(Matrix::nearPD(mat)$mat)`.

Value

N x N covariance matrix

Examples

```
A <- array(rnorm(4 * 2 * 3), dim = c(2, 3, 4))
A.cvm <- dotcvm(A)
```

`mirrorfill` *Fill missing symmetrical landmarks for all specimens in an array using mirrored values from other side of a bilaterally symmetrical object where present*

Description

Given an N x 3 x M matrix, where N is the number of landmarks, 3 is the number of dimensions, and M is the number of specimens, fill in missing landmarks using their mirrored counterpart.

Usage

```
mirrorfill(A, l1, l2)
```

Arguments

A	An N x 3 x M matrix where N is the number of landmarks, 3 is the number of dimensions, and M is the number of specimens.
l1	Vector of indices for which landmarks to use to make a specimen midline
l2	Vector or matrix of pairs of symmetrical landmarks

Details

l2 should be either

- An even length vector containing pairs of landmarks on either side of the specimen. i.e. l2[1] and l2[2] are paired, l2[3] and l2[4] are paired etc.
- A two column matrix with each row giving a pair of symmetrical landmarks.

l2 should be an even number length containing pairs of landmarks on either side of the specimen.

Examples

```
# Create array
A <- array(rep(1:36, by = 4), dim = c(12, 3, 4))

# Make it symmetrical
A[7:12, 1:2, ] <- A[1:6, 1:2, ]
A[7:12, 3, ] <- -A[1:6, 3, ]

# Remove some data points
missinga <- A
missinga[1:2, , 1:3] <- NA

mirrorA <- mirrorfill(missinga, l1 = c(3:6, 9:12), l2 = c(1, 7, 2, 8))
```

---

mirrorfill1 *Fill missing landmarks for a single specimen using mirrored values from other side of object*

Description

Given an n x 3 matrix, replace a set of landmarks using their mirrored counterpart.

Usage

```
mirrorfill1(s, l1, l2)
```

Arguments

s	An n x 3 matrix containing 3D landmark data of n landmarks.
l1	Vector of indices for which landmarks to use to make a specimen midline.
l2	Vector or matrix of pairs of symmetrical landmarks.

Details

l2 should be either

- An even length vector containing pairs of landmarks on either side of the specimen. i.e. l2[1] and l2[2] are paired, l2[3] and l2[4] are paired etc.
- A two column matrix with each row giving a pair of symmetrical landmarks.

Examples

```
# Make data that is reflected in x plane
s <- matrix(rep(1:21, 2), byrow = TRUE, ncol = 3)
s[1:7, 1] <- -s[1:7, 1]

# Now remove some data
s[1, ] <- NA

# Mirror point 1 using it's complimentary landmark, point 8.
mirrorS <- mirrorfill1(s, l1 = c(2:7, 9:14), l2 = c(1, 8))
```

---

plotSpecimens *Plot an array of specimen landmark data in an interactive 3D frame*

Description

This function requires the rgl package. Given a N x 3 x M array (where M is the number of specimens and N is the number of landmarks), as used elsewhere in this package, plot each specimen in a different colour in an intereactive 3D frame.

Usage

```
plotSpecimens(A, l1 = NULL, midlineSpecimens = NULL, cols = NULL,
  bySpecimen = TRUE, planeOptions = NULL, ...)
```

Arguments

A	An N x 3 x M array.
l1	Optional vector of indices for which landmarks to use to make a specimen midline. If NULL, no midline plane is plotted.
midlineSpecimens	Numeric vector indicating which specimens should be used to built the midline plane. If NULL, but l1 is defined, all specimens are used.
cols	A vector of colours.
bySpecimen	Logical that determined whether points should be coloured by specimen (default) or by landmark.
planeOptions	Named list of parameters passed to rgl.material to control the appearence of plotted mirror planes.
...	Further parameters passed to plot3d.

See Also

plot3d mirrorfill planes3d rgl.material

Examples

```
A <- array(rep(rnorm(3 * 20, sd = 30), by = 6) + rnorm(6 * 20 * 3),
       dim = c(20, 3, 6))
plotSpecimens(A)

plotSpecimens(A, bySpecimen = FALSE)

plotSpecimens(A, cols = grey(seq(0, 1, length.out = 6)))

plotSpecimens(A, l1 = c(1:4), planeOptions = list(alpha = 0.4, color = 'red'))
```

`procrustes` *Conducts Procrustes superimposition to align 3D shapes with or without scaling to centroid size.*

Description

Conducts Procrustes superimposition to align 3D shapes with or without scaling to centroid size. Skips any missing values in computation of Procrustes coordinates.

Usage

```
procrustes(A, scale = TRUE, scaleDelta = FALSE, maxiter = 1000,
  tolerance = 1e-05)
```

Arguments

`A`	N x 3 x M matrix where N is the number of landmarks, 3 is the number of dimensions, and M is the number of specimens
`scale`	Logical indicating whether objects should be scaled to unit centroid size
`scaleDelta`	Logical determining whether deltaa should be scaled by the total number of landmarks.
`maxiter`	Maximum number of iterations to attempt
`tolerance`	Difference between two iterations that will cause the search to stop.

Details

A number of computations are run until the difference between two iterations is less than `tolerance`. The more specimens and landmarks you have, the less each landmark is allowed to move before this tolerance is reached. Setting `scaleDelta = TRUE` will make the alignment run faster but have potentially less well aligned results. But the alignment between a large and small array of shapes should be more comparable with `scaleDelta = TRUE`. However, preliminary tests imply that run time scales linearly with `scaleDelta` set to `TRUE` or `FALSE`.

Value

A new (N x 3 x M) array, where each 3d vector has been rotated and translated to minimize distances among specimens, and scaled to unit centroid size if requested.

Examples

```
# Make an array with 6 specimens and 20 landmarks
A <- array(rep(rnorm(6 * 20, sd = 20), each = 6) + rnorm(20 * 3 * 6 ),
      dim = c(20, 3, 6))

# Align the data (although it is already largely aligned)
aligned <- procrustes(A)

plotSpecimens(aligned)
```

Index