Package 'geomorph'

July 19, 2014

Date 2014-07-14

Type Package

Title Geometric morphometric analyses of 2d/3d landmark data

Version 2.1.1

Author Dean Adams, Michael Collyer, Erik Otarola-Castillo, Emma Sherratt

Maintainer Dean Adams <dcadams@iastate.edu>

Depends R (>= 2.10), rgl

Imports jpeg, vegan, ape, geiger, phytools

Description Read, manipulate, and digitize landmark data, generate shape variables via Procrustes analysis for points, curves and surfaces, perform shape analyses, and provide graphical depictions of shapes and patterns of shape variation

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-07-19 08:41:30

R topics documented:

geomorph-package *Geometric morphometric analyses for 2D/3D data*

Description

Geometric morphometric analyses for 2D/3D data

Author(s)

Dean C. Adams, Michael Collyer, Erik Otarola-Castillo, & Emma Sherratt

Functions in this package allow one to read, manipulate, and digitize landmark data; generate shape variables via Procrustes analysis for points, curves and surface data, perform statistical analyses of shape variation and covariation, and provide graphical depictions of shapes and patterns of shape variation.

arrayspecs *Convert landmark data matrix into array (p x k x n)*

Description

Convert a matrix of landmark coordinates into a 3-dimensional array

Usage

```
arrayspecs(A, p, k)
```

Arguments

A	A matrix containing landmark coordinates for a set of specimens
p	Number of landmarks
k	Number of dimensions (2 or 3)

Details

This function converts a matrix of landmark coordinates into a (p x k x n) array, which is the required input format for many functions in geomorph. The input matrix can be arranged such that the coordinates of each landmark are found on a separate row, or that each row contains all landmark coordinates for a single specimen.

Value

Function returns a (p x k x n) array, where p is the number of landmark points, k is the number of landmark dimensions (2 or 3), and n is the number of specimens. The third dimension of this array contains names for each specimen if specified in the original input matrix.

Author(s)

Dean Adams & Mike Collyer

Examples

```
x<-matrix(rnorm(18),nrow=3)  # Random triangles (all coordinates on same row for each triangle)
arrayspecs(x,3,2)

x2<-matrix(rnorm(18),ncol=2) # Random triangles (each landmark on its own row)
arrayspecs(x2,3,2)
```

`bilat.symmetry` *Analysis of bilateral symmetry*

Description

Function performs an analysis of directional and fluctuating asymmetry for bilaterally symmetric objects

Usage

```
bilat.symmetry(A, ind = NULL, side = NULL, replicate = NULL,
  object.sym = FALSE, land.pairs = NULL, warpgrids = TRUE, mesh = NULL,
  verbose = FALSE)
```

Arguments

`A`	An array (p x k x n) containing GPA-aligned coordinates for a set of specimens [for "object.sym=FALSE, A is of dimension (n x k x 2n)]
`ind`	A vector containing labels for each individual. For matching symmetry, the matched pairs receive the same label (replicates also receive the same label).
`side`	An optional vector (for matching symmetry) designating which object belongs to which 'side-group'
`replicate`	An optional vector designating which objects belong to which group of replicates
`object.sym`	A logical value specifying whether the analysis should proceed based on object symmetry =TRUE or matching symmetry =FALSE
`land.pairs`	An optional matrix (for object symmetry) containing numbers for matched pairs of landmarks across the line of symmetry
`warpgrids`	A logical value indicating whether deformation grids for directional and fluctuating components of asymmetry
`mesh`	A mesh3d object to be warped to represent shape deformation of the directional and fluctuating components of asymmetry if warpgrids= TRUE (see `warpRefMesh`).
`verbose`	A logical value indicating whether the output is basic or verbose (see Value below)

Details

The function quantifies components of shape variation for a set of specimens as described by their patterns of symmetry and asymmetry. Here, shape variation is decomposed into variation among individuals, variation among sides (directional asymmetry), and variation due to an individual x side interaction (fluctuating symmetry). These components are then statistically evaluated using Procrustes ANOVA and Goodall's F tests (i.e. an isotropic model of shape variation). Methods for both matching symmetry and object symmetry can be implemented. Matching symmetry is when each object contains mirrored pairs of structures (e.g., right and left hands) while object symmetry is when a single object is symmetric about a midline (e.g., right and left sides of human faces). Analytical and computational details concerning the analysis of symmetry in geometric morphometrics can be found in Mardia et al. 2000; Klingenberg et al. 2002.

Analyses of symmetry for matched pairs of objects is implemented when object.sym=FALSE. Here, a 3D array [p x k x 2n] contains the landmark coordinates for all pairs of structures (2 structures for each of n specimens). Because the two sets of structures are on opposite sides, they represent mirror images, and one set must be reflected prior to the analysis to allow landmark correspondence. IT IS ASSUMED THAT THE USER HAS DONE THIS PRIOR TO PERFORMING THE SYMMETRY ANALYSIS. Reflecting a set of specimens may be accomplished by multiplying one coordinate dimension by '-1' for these structures (either the x-, the y-, or the z-dimension). A vector containing information on individuals and sides must also be supplied. Replicates of each specimen may also be included in the dataset, and when specified will be used as measurement error (see Klingenberg and McIntyre 1998).

Analyses of object symmetry is implemented when object.sym=TRUE. Here, a 3D array [p x k x n] contains the landmark coordinates for all n specimens. To obtain information about asymmetry, the function generates a second set of objects by reflecting them about one of their coordinate axes. The landmarks across the line of symmetry are then relabeled to obtain landmark correspondence. The user must supply a list of landmark pairs. A vector containing information on individuals must also be supplied. Replicates of each specimen may also be included in the dataset, and when specified will be used as measurement error.

Value

Function returns a list with the following components:

`ANOVA.shape`	Procrustes ANOVA table assessing patterns of shape asymmetry
`ANOVA.size`	Procrustes ANOVA table assessing patterns of shape asymmetry (when object.sym=FALSE)
`symm.shape`	The symmetric component of shape variation of the aligned specimens (when verbose=TRUE)
`asymm.shape`	The asymmetric component of shape variation of the aligned specimens (when verbose=TRUE)

Author(s)

Dean Adams & Emma Sherratt

References

Klingenberg, C.P. and G.S. McIntyre. 1998. Quantitative genetics of geometric shape in the mouse mandible. Evolution. 55:2342-2352.

Mardia, K.V., F.L. Bookstein, and I.J. Moreton. 2000. Statistical assessment of bilateral symmetry of shapes. Biometrika. 87:285-300.

Klingenberg, C.P., M. Barluenga, and A. Meyer. 2002. Shape analysis of symmetric structures: quantifying variation among individuals and asymmetry. Evolution. 56:1909-1920.

Examples

```
#Example of matching symmetry

data(mosquito)
bilat.symmetry(mosquito$wingshape,ind=mosquito$ind,side=mosquito$side,
replicate=mosquito$replicate,object.sym=FALSE)

#Example of object symmetry

data(scallops)
bilat.symmetry(scallops$coorddata,ind=scallops$ind,object.sym=TRUE,land.pairs=scallops$land.pairs)
```

---

buildtemplate *Build 3D surface template*

Description

An interactive function to build template of three-dimensional (3D) surface sliding semilandmarks. Input for the function is either a matrix of vertex coordinates defining a 3D surface object or a mesh3d object as obtained from `read.ply`.

Usage

```
buildtemplate(spec, fixed, surface.sliders, ptsize = 1, center = TRUE)
```

Arguments

`spec`	Name of surface file, as either an object of class shape3d/mesh3d, or matrix of three-dimensional vertex coordinates.
`fixed`	numeric: The number of fixed template landmarks
`surface.sliders`	numeric: The number of template surface sliders desired
`ptsize`	numeric: Size to plot the mesh points (vertices), e.g. 0.1 for dense meshes, 3 for sparse meshes
`center`	Logical Whether the object 'spec' should be centered prior to digitizing (default center=TRUE)

Details

Function constructs a template of fixed landmarks and n "surface sliders", semilandmarks that slide over a surface. The user digitizes the fixed points (see digitizing below), then the function finds n surface semilandmarks following algorithm outlined in Gunz et al. (2005) and Mitteroecker and Gunz (2009). Surface semilandmarks are roughly equidistant set of predetermined number of points, chosen over the mesh automatically using a nearest-neighbor approach. The set of fixed and surface slider landmarks are exported as a "template", which is used to extract a set of similarly numbered landmarks on every specimen using function `digitsurface`. Some of the "fixed" landmarks can be later designated as "curve sliders" using function `define.sliders.3d` if required - see details in `digit.fixed`. Because template matching is based on the correspondence of fixed landmark points in the template and the specimen, a minimum of four fixed landmarks must be used.

To ensure a strong match between the scan and the template, it is recommended that a reasonable number of fixed points be used. These fixed points can be designated as "curve sliders" later using function `define.sliders.3d`, see the function `digit.fixed` for details. NOTE: Function centers the mesh before digitizing by default (center=TRUE). If one chooses not to center, specimen may be difficult to manipulate in rgl window.

Digitizing: Digitizing using buildtemplate is interactive between landmark selection using a mouse (see below for instructions), and the R console. Once a point is selected, the user is asked if the system should keep or discard the selection #'(y/n). If "y", the user is asked to continue to select the next landmark. If "n" the removes the last chosen landmark, and the user is asked to select it again. This can be repeated until the user is comfortable with the landmark chosen.

To digitize with a standard 3-button (PC):

1. the RIGHT mouse button (primary) to select points to be digitized (click-drag a box around a vertex to select as landmark)
2. the LEFT mouse button (secondary) is used to rotate mesh,
3. the mouse SCROLLER (third/middle) is used to zoom in and out.

NOTE: Digitizing functions on MACINTOSH computers using a standard 3-button mice works as specified. Macs using platform specific single button mice, XQuartz must be configured: go to Preferences > Input > tick "Emulate three button mouse":

1. press button to rotate 3D mesh,
2. press button while pressing COMMAND key to select points to be digitized (click-drag a box around a vertex to select as landmark),
3. press button while pressing OPTION key to adjust mesh perspective.
4. the mouse SCROLLER or trackpad two finger scroll is used to zoom in an out.

NOTE: there is no pan (translate) functionality in rgl library for all platforms at this time. The template can be edited using function `editTemplate`.

Value

Function writes to the working directory three files: an NTS file with the name of the specimen and .nts suffix containing the landmark coordinates, "template.txt" containing the same coordinates for use with the function `digitsurface`, and "surfslide.csv", a file containing the address of the landmarks defined as "surface sliders" for use with `gpagen`. Function also returns to console an n x 3 matrix containing the x,y,z coordinates of the digitized landmarks.

Author(s)

Erik Otarola-Castillo & Emma Sherratt

References

Gunz P, Mitteroecker P, & Bookstein FJ (2005) Semilandmarks in Three Dimensions. Modern Morphometrics in Physical Anthropology, ed Slice DE (Springer-Verlag, New York), pp 73-98.

Mitteroecker P & Gunz P (2009) Advances in Geometric Morphometrics. Evolutionary Biology 36(2):235-247.

See Also

`read.ply`

`digit.fixed`

`digitsurface`

`compare.evol.rates` *Comparing rates of shape evolution on phylogenies*

Description

Function calculates rates of shape evolution for two or more groups of species on a phylogeny from a set of Procrustes-aligned specimens

Usage

```
compare.evol.rates(phy, A, gp, iter = 999)
```

Arguments

`phy`	A phylogenetic tree of class phylo - see `read.tree` in library ape
`A`	A matrix (n x [p x k]) or 3D array (p x k x n) containing GPA-aligned coordinates for a set of specimens
`gp`	A factor array designating group membership
`iter`	Number of iterations for significance testing

Details

The function compares rates of morphological evolution for two or more groups of species on a phylogeny, under a Brownian motion model of evolution. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with `gpagen`]. The approach is based on the distances between species in morphospace after phylogenetic transformation (Adams 2014). From the data the rate of shape evolution for each group is calculated, and a ratio of rates is obtained. If three or more groups of species are used, the ratio of the maximum to minimum rate is used as a test statistic (see Adams 2014). Significance testing is accomplished by phylogenetic simulation in which tips data are obtained under Brownian motion using a single evolutionary

rate for all species on the phylogeny. If three or more groups of species are used, pairwise p-values are also returned. A histogram of evolutionary rate ratios obtained via phylogenetic simulation is presented, with the observed value designated by an arrow in the plot. The function can be used to obtain a rate for the whole dataset of species by using a dummy group factor assigning all species to one group.

This function can be used with univariate data (i.e. centroid size) if imported as matrix with row-names giving the taxa names.

Value

Function returns a list with the following components:

`sigma.d`	The phylogenetic evolutionary rate for all species on the phylogeny
`sigmad.all`	The phylogenetic evolutionary rate for each group of species on the phylogeny
`sigmad.ratio`	The ratio of maximum to minimum evolutionary rates
`pvalue`	The significance level of the observed ratio
`pairwise.pvalue`	Matrix of pairwise significance levels comaring each pair of rates

Author(s)

Dean Adams & Emma Sherratt

References

Adams, D.C. 2014. Quantifying and comparing phylogenetic evolutionary rates for shape and other high-dimensional phenotypic data. Syst. Biol. 63:166-177.

Examples

```
data(plethspecies)
Y.gpa<-gpagen(plethspecies$land)    #GPA-alignment

 gp.end<-factor(c(0,0,1,0,0,1,1,0,0))  #endangered species vs. rest
 names(gp.end)<-plethspecies$phy$tip

#Calculate rates of shape
compare.evol.rates(plethspecies$phy,Y.gpa$coords,gp=gp.end,iter=49)

#Calculate rates of size
Csize <- matrix(Y.gpa$Csize, dimnames=list(names(Y.gpa$Csize))) # make matrix Csize with names
compare.evol.rates(plethspecies$phy,Csize,gp=gp.end,iter=49)
```

`compare.modular.partitions` *Compare modular signal to alternative landmark subsets*

Description

Function quantifies the degree of morphological integration between two or more modules of Procrustes-aligned landmark coordinates and compares this to patterns found by randomly assigning landmarks into subsets

Usage

```
compare.modular.partitions(A, partition.gp, iter = 999)
```

Arguments

`A`	A 3D array (p x k x n) containing GPA-aligned coordinates for all specimens, or a matrix (n x variables)
`partition.gp`	A list of which landmarks (or variables) belong in which partition (e.g. A,A,A,B,B,B,C,C,C)
`iter`	Number of iterations for significance testing

Details

The function quantifies the degree of morphological integration between two or more modules of shape data as defined by landmark coordinates, and compares this to modular signals found by randomly assigning landmarks to modules. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with `gpagen`]. The degree of morphological integration is quantified using the RV coefficient (Klingenberg 2009). If more than two modules are defined, the average RV coefficient is utilized (see Klingenberg 2009). The RV coefficient for the observed modular hypothesis is then compared to a distribution of values obtained by randomly assigning landmarks into subsets, with the restriction that the number of landmarks in each subset is identical to that observed in each of the original partitions. A significant modular signal is found when the observed RV coefficient is small relative to this distribution (see Klingenberg 2009). A histogram of coefficients obtained via resampling is presented, with the observed value designated by an arrow in the plot.

Landmark groups can be defined using `define.modules`, or made by hand (see example below). To use this method with other data (i.e., a set of length measurements), the input A should be a matrix of n rows of specimens and variables arranged in columns. In this case, the partition.gp input should have each variable assigned to a partition.

Value

Function returns a list with the following components:

`RV`	The estimate of morphological integration
`pvalue`	The significance level of the observed signal

RV.min	The minimal RV coefficient found via landmark permutation
RV.min.partitions	A list of landmarks assigned to partitions that yields the minimal RV coefficient

Author(s)

Dean Adams

References

Klingenberg, C. P. 2009. Morphometric integration and modularity in configurations of landmarks: tools for evaluating a priori hypotheses. Evol. Develop. 11:405-421.

Examples

```
data(plethodon)
Y.gpa<-gpagen(plethodon$land)    #GPA-alignment
 #landmarks on the skull and mandible assigned to partitions
land.gps<-c("A","A","A","A","A","B","B","B","B","B","B","B")

compare.modular.partitions(Y.gpa$coords,land.gps,iter=99)
#Result implies that the skull and mandible are not independent modules
```

---

define.modules *Define modules (landmark partitions)*

Description

An interactive function to define which landmarks should be assigned to each module (landmark partition).

Usage

```
define.modules(spec, nmodules)
```

Arguments

spec	Name of specimen, as an object matrix containing 2D landmark coordinates
nmodules	Number of modules to be defined

Details

Function takes a matrix of two-dimensional digitized landmark coordinates and allows the user to assign landmarks to each module. The output is a list of which landmarks belong in which partition, to be used by compare.modular.partitions. The number of modules is chosen by the user (up to five).

Selection: Choosing which landmarks will be included in each module involves landmark selection using a mouse in the plot window. The user is prompted to select each landmarks for module 1: using the LEFT mouse button (or regular button for Mac users), click on the hollow circle to choose the landmark. Selected landmarks will be filled in. When all landmarks for module 1 are chosen, press 'esc', and then start selecting landmarks for module 2. Repeat until all modules are defined.

Note: Function currently only implemented for 2D landmark data.

Value

Function returns a vector of which landmarks belong in which module (e.g. A,A,A,B,B,B,C,C,C) to be used with `compare.modular.partitions` option 'landgroups'.

Author(s)

Emma Sherratt

See Also

`compare.modular.partitions`

`define.sliders.2d` *Select points to "slide" along two-dimensional curves.*

Description

An interactive function to define which landmarks will "slide" along two-dimensional (2D) curves.

Usage

```
define.sliders.2d(spec, nsliders)
```

Arguments

`spec`	Name of specimen, as an object matrix containing 2D landmark coordinates
`nsliders`	Number of landmarks to be semilandmarks that slide along curves

Details

Function takes a matrix of digitized landmark coordinates, such as made by `digitize2d`, and helps user choose which landmarks will be treated as "curve sliders" in Generalized Procrustes analysis `gpagen`. This type of semilandmark "slides" along curves lacking known landmarks (see Bookstein 1997 for algorithm details). Each sliding semilandmark ("sliders") will slide between two designated points, along a line tangent to the specified curvature.

Selection: Choosing which landmarks will be sliders involves landmark selection using a mouse in the plot window. To define the sliders, for each sliding landmark along the curve in the format 'before-slider-after', using the LEFT mouse button (or regular button for Mac users), click on the hollow circle to choose the landmark in the following order:

1. Click to choose the first landmark between which semi-landmark will "slide",
2. Click to choose sliding landmark,
3. Click to choose the last landmark between which semi-landmark will "slide", Selected landmarks will be filled in and lines are drawn connecting the three landmarks, and will highlight the sliding semilandmark in red and the flanking landmarks in blue.

This procedure is overlapping, so for example, a curve defined by a sequence of semilandmarks, the user must select the 2nd point of the first three to be the 1st for the next e.g. 1 2 3 then 2 3 4, then 3 4 5 etc.

Value

Function returns a 'nsliders-x-3' matrix containing the landmark address of the curve sliders, indicating the landmarks between which the slider landmarks will "slide". The matrix is also written to working directory as "curveslide.csv". Matrix (or "curveslide.csv") is designed for use by gpagen during GPA.

Author(s)

Dean Adams, Erik Otarola-Castillo, Emma Sherratt

References

Bookstein, F. J. 1997 Landmark Methods for Forms without Landmarks: Morphometrics of Group Differences in Outline Shape. Medical Image Analysis 1(3):225-243.

See Also

digitize2d, gpagen

`define.sliders.3d` *Choose points to "slide" along three-dimensional curves.*

Description

An interactive function to define which digitized landmarks of an '*.nts' file will "slide" along three-dimensional (3D) curves.

Usage

```
define.sliders.3d(spec, nsliders, surfsliders = FALSE)
```

Arguments

spec	Name of specimen, as an object matrix containing 3D landmark coordinates
nsliders	Number of landmarks to be semilandmarks that slide along curves
surfsliders	Logical If spec contains landmarks that are "surface sliders", made by buildtemplate, "surfslide.csv" should be in working directory

Details

Function takes a matrix of digitized landmark coordinates, such as made by digit.fixed, or buildtemplate and helps user choose which landmarks will be treated as "curve sliders" in Generalized Procrustes analysis gpagen. This type of semilandmark "slides" along curves lacking known landmarks (see Bookstein 1997 for algorithm details). Each sliding semilandmark ("sliders") will slide between two designated points, along a line tangent to the specified curvature.

Selection: Choosing which landmarks will be sliders involves landmark selection using a mouse in the rgl plot window. With a standard 3-button (PC) buildtemplate uses:

1. the RIGHT mouse button (primary) to choose points to be defined as sliders (click-drag a box around a vertex to select),
2. the LEFT mouse button (secondary) is used to rotate mesh,
3. the mouse SCROLLER (third/middle) is used to zoom in and out.

NOTE: Digitizing functions on MACINTOSH computers using a standard 3-button mice works as specified. Macs using platform specific single button mice, XQuartz must be configured: go to Preferences > Input > tick "Emulate three button mouse":

1. press button to rotate 3D mesh,
2. press button while pressing COMMAND key to select points to be digitized (click-drag a box around a vertex to select as landmark),
3. press button while pressing OPTION key to adjust mesh perspective.
4. the mouse SCROLLER or trackpad two finger scroll is used to zoom in an out.

To define the sliders, for each sliding landmark along the curve in the format 'before-slider-after':

1. Click-drag to choose the first landmark between which semi-landmark will "slide",
2. Click-drag to choose sliding landmark,
3. Click-drag to choose the last landmark between which semi-landmark will "slide", Screen will show lines connecting the three landmarks, and will highlight the sliding semilandmark in red.

This procedure is overlapping, so for example a curve defined by a sequence of semilandmarks, the user must select the 2nd point of the first three to be the 1st for the next e.g. 1 2 3 then 2 3 4, etc.

Value

Function returns a 'curves x 3' matrix containing the landmark address of the curve sliders, indicating the points between which the selected point will "slide". Written to the working directory is this matrix as "curveslide.csv". Matrix (or "curveslide.csv") is designed for use by gpagen during GPA.

Author(s)

Erik Otarola-Castillo & Emma Sherratt

References

Bookstein, F. J. 1997 Landmark Methods for Forms without Landmarks: Morphometrics of Group Differences in Outline Shape. Medical Image Analysis 1(3):225-243.

See Also

digit.fixed, digitsurface, gpagen

---

digit.fixed *Digitize 3D landmarks*

Description

An interactive function to digitize three-dimensional (3D) landmarks. Input for the function is either a matrix of vertex coordinates defining a 3D surface object or a mesh3d object as obtained from read.ply.

Usage

```
digit.fixed(spec, fixed, index = FALSE, ptsize = 1, center = TRUE)
```

Arguments

spec	An object of class shape3d/mesh3d, or matrix of 3D vertex coordinates
fixed	Numeric The number landmarks (fixed, and curve sliders if desired)
index	Logical Whether selected landmark addresses should be returned (internal use only)
ptsize	Numeric Size to plot the mesh points (vertices), e.g. 0.1 for dense meshes, 3 for sparse meshes
center	Logical Whether the object 'spec' should be centered prior to digitizing (default center=TRUE)

Details

Function for digitizing "n" three-dimensional landmarks. The landmarks are "fixed" (traditional landmarks). They can be later designated as "curve sliders" (semilandmarks, that will "slide" along curves lacking known landmarks if required. A sliding semi-landmark ("sliders") will slide between two designated points, along a line tangent to the specified curvature, and must be defined as "sliders" using function define.sliders.3d or with similar format matrix made outside R.

For 3D "surface sliders" (surface semilandmarks that slide over a surface) the function digitsurface should be used instead. NOTE: Function centers the mesh before digitizing by default (center=TRUE). If one chooses not to center, specimen may be difficult to manipulate in rgl window.

Digitizing: Digitizing is interactive between landmark selection using a mouse (see below for instructions), and the R console. Once a point is selected, the user is asked if the system should keep or discard the selection #'(y/n). If "y", the user is asked to continue to select the next landmark. If "n" the removes the last chosen landmark, and the user is asked to select it again. This can be repeated until the user is comfortable with the landmark chosen.

To digitize with a standard 3-button (PC):

1. the RIGHT mouse button (primary) to select points to be digitized (click-drag a box around a vertex to select as landmark),
2. the LEFT mouse button (secondary) is used to rotate mesh,
3. the mouse SCROLLER (third/middle) is used to zoom in and out.

NOTE: Digitizing functions on MACINTOSH computers using a standard 3-button mice works as specified. Macs using platform specific single button mice, XQuartz must be configured: go to Preferences > Input > tick "Emulate three button mouse":

1. press button to rotate 3D mesh,
2. press button while pressing COMMAND key to select points to be digitized (click-drag a box around a vertex to select as landmark),
3. press button while pressing OPTION key to adjust mesh perspective.
4. the mouse SCROLLER or trackpad two finger scroll is used to zoom in an out.

NOTE: there is no pan (translate) functionality in rgl library for all platforms at this time.

Value

Function writes to the working directory an NTS file with the name of the specimen and .nts suffix containing the landmark coordinates. If index=FALSE function returns to the console an n x 3 matrix containing the x,y,z coordinates of the digitized landmarks. If index=TRUE, function returns a list:

`selected`	a matrix containing the x,y,z coordinates of the digitized landmarks
`fix`	a matrix of addresses for landmarks that are "fixed" (for internal use)

Author(s)

Erik Otarola-Castillo & Emma Sherratt

See Also

`read.ply`

`digitize2d` *Digitize 2D landmarks.*

Description

An interactive function to digitize two-dimensional(2D) landmarks from .jpg files.

Usage

```
digitize2d(filelist, nlandmarks, scale = NULL, tpsfile, verbose = TRUE)
```

Arguments

filelist	A list of names of jpeg images to be digitized.
nlandmarks	Number of landmarks to be digitized.
scale	An optional vector containing the length of the scale to be placed on each image.
tpsfile	The name of a TPS file to be created or read
verbose	logical. User decides whether to digitize in verbose or silent format (see details), default is verbose

Details

This function may be used for digitizing 2D landmarks from jpeg images (.jpg). The user provides a list of image names, the number of landmarks to be digitized, and the name of an output TPS file. An option is included to allow the user to digitize a scale on each image to convert the landmark coordinates from pixels into meaningful units. Landmarks to be digitized can include both fixed landmarks and semi-landmarks, the latter of which are to be designated as "sliders" for subsequent analysis (see the function `define.sliders.2d`).

The Digitizing Session: Users may digitize all specimens in one session, or may return at a later time to complete digitizing. In the latter case, the user provides the same filelist and TPS file and the function will determine where the user left off.

If specimens have missing landmarks, these can be incorporated during the digitizing process using the 'a' option as described below (a=absent).

Specimen Digitizing: Digitizing landmarks involves landmark selection using a mouse in the plot window, using the LEFT mouse button (or regular button for Mac users):

1. Digitize the scale bar (if requested) by selecting the two end points. Use a single click for start and end points. The user is asked whether the system should keep or discard the digitized scale bar.
2. Digitize each landmark with single click and the landmark is shown in red.

If verbose = TRUE, digitizing is interactive between landmark selection using a mouse and the R console. Once a landmark is selected, the user is asked if the system should keep or discard the selection (y/n/a). If "y", the user is asked to continue to select the next landmark. If "n", the user is asked to select it again.

To digitize a missing landmark, simply click on any location in the image. Then, when prompted to keep selection, choose 'a' (for absent). Missing landmarks can only be included during the digitizing process when verbose=TRUE.

If verbose = FALSE the digitizing of landmarks is continuous and uninterupted. Here the user will not be prompted to approve each landmark selection.

At the end of digitizing, the landmark coordinates are written to a TPS file. The x,y values are scaled if a vector fo scales is included."

Value

Function returns a tps file containing the digitized landmark coordinates.

Author(s)

Dean Adams Erik Otarola-Castillo and Emma Sherratt

`digitsurface` *Digitize 3D fixed landmarks and surface semilandmarks.*

Description

An interactive function to digitize three-dimensional (3D) landmarks on a surface lacking known landmarks. Input for the function is either a matrix of vertex coordinates defining a 3D surface object or a mesh3d object as obtained from `read.ply`.

Usage

```
digitsurface(spec, fixed, ptsize = 1, center = TRUE)
```

Arguments

`spec`	Name of surface file, as either an object of class shape3d/mesh3d, or matrix of three-dimensional vertex coordinates.
`fixed`	numeric: The number of fixed template landmarks
`ptsize`	numeric: Size to plot the mesh points (vertices), e.g. 0.1 for dense meshes, 3 for sparse meshes
`center`	Logical Whether the object 'spec' should be centered prior to digitizing (default center=TRUE)

Details

Function for digitizing fixed 3D landmarks and placing "surface sliders", semilandmarks that slide over a surface. Following selection of fixed points (see digitizing below), function finds surface semilandmarks following algorithm outlined in Gunz et al. (2005) and Mitteroecker and Gunz (2009). digitsurface finds the same number of surface semilandmarks as the template (created by `buildtemplate`) by downsampling scanned mesh, registering template with current specimen via GPA. A nearest neighbor algorithm is used to match template surface landmarks to current specimen's. To use function digitsurface, the template must be constructed first, and 'template.txt' be in the working directory. Because template matching is based on the correspondence of fixed landmark points in the template and the specimen, a minimum of four fixed landmarks must be used.

Some of the "fixed" landmarks digitized with digitsurface can be later designated as "curve sliders" using function `define.sliders.3d` if required (see details in `digit.fixed`). NOTE: Function centers the mesh before digitizing by default (center=TRUE). If one chooses not to center, specimen may be difficult to manipulate in rgl window.

Digitizing: Digitizing is interactive between landmark selection using a mouse (see below for instructions), and the R console. Once a point is selected, the user is asked if the system should keep or discard the selection #'(y/n). If "y", the user is asked to continue to select the next landmark. If "n" the removes the last chosen landmark, and the user is asked to select it again. This can be repeated until the user is comfortable with the landmark chosen.

To digitize with a standard 3-button (PC):

1. the RIGHT mouse button (primary) to select points to be digitized (click-drag a box around a vertex to select as landmark),
2. the LEFT mouse button (secondary) is used to rotate mesh,
3. the mouse SCROLLER (third/middle) is used to zoom in and out.

NOTE: Digitizing functions on MACINTOSH computers using a standard 3-button mice works as specified. Macs using platform specific single button mice, XQuartz must be configured: go to Preferences > Input > tick "Emulate three button mouse":

1. press button to rotate 3D mesh,
2. press button while pressing COMMAND key to select points to be digitized (click-drag a box around a vertex to select as landmark),
3. press button while pressing OPTION key to adjust mesh perspective.
4. the mouse SCROLLER or trackpad two finger scroll is used to zoom in an out.

NOTE: there is no pan (translate) functionality in rgl library for all platforms at this time.

Value

Function writes to the working directory an NTS file with the name of the specimen and .nts suffix containing the landmark coordinates.

Author(s)

Erik Otarola-Castillo & Emma Sherratt

References

Gunz P, Mitteroecker P, & Bookstein FJ (2005) Semilandmarks in Three Dimensions. Modern Morphometrics in Physical Anthropology, ed Slice DE (Springer-Verlag, New York), pp 73-98.

Mitteroecker P & Gunz P (2009) Advances in Geometric Morphometrics. Evolutionary Biology 36(2):235-247.

See Also

`buildtemplate`

`read.ply`

`digit.fixed`

`editTemplate` *Edit 3D template*

Description

An interactive function to remove landmarks from a 3D template file.

Usage

```
editTemplate(template, fixed, n)
```

Arguments

`template`	Matrix of template 3D coordinates.
`fixed`	Number of "fixed" landmark points (non surface sliding points)
`n`	Number of points to be removed

Details

Function edits a 'template.txt' file made by `buildtemplate`, which must be in current working directory. Function overwrites 'template.txt' in working directory with edited version. Use read.table("template.txt", header = T).

Selection: Choosing which landmarks will be deleted involves landmark selection using a mouse in the rgl plot window. With a standard 3-button (PC) buildtemplate uses:

1. the RIGHT mouse button (primary) to choose points to be deleted (click-drag a box around landmark),
2. the LEFT mouse button (secondary) is used to rotate mesh,
3. the mouse SCROLLER (third/middle) is used to zoom in and out.

NOTE: Digitizing functions on MACINTOSH computers using a standard 3-button mice works as specified. Macs using platform specific single button mice, XQuartz must be configured: go to Preferences > Input > tick "Emulate three button mouse":

1. press button to rotate 3D mesh,
2. press button while pressing COMMAND key to select points to be digitized (click-drag a box around a vertex to select as landmark),
3. press button while pressing OPTION key to adjust mesh perspective.
4. the mouse SCROLLER or trackpad two finger scroll is used to zoom in an out.

Value

Function returns a matrix containing the x,y,z coordinates of the new template landmarks. Function also writes to working directory 'template.txt' containing the x,y,z coordinates of the template

Author(s)

Erik Otarola-Castillo & Emma Sherratt

`estimate.missing` *Estimate locations of missing landmarks using the thin-plate spline*

Description

A function for estimating the locations of missing landmarks

Usage

```
estimate.missing(A, method = c("TPS", "Reg"))
```

Arguments

`A`	An array (p x k x n) containing landmark coordinates for a set of specimens
`method`	Method for estimating missing landmark locations

Details

The function estimates the locations of missing landmarks for incomplete specimens in a set of landmark configurations, where missing landmarks in the incomplete specimens are designated by NA in place of the x,y,z coordinates. Two distinct approaches are implemented.

The first approach (method="TPS") uses the thin-plate spline to interpolate landmarks on a reference specimen to estimate the locations of missing landmarks on a target specimen. Here, a reference specimen is obtained from the set of specimens for which all landmarks are present, Next, each incomplete specimen is aligned to the reference using the set of landmarks common to both. Finally, the thin-plate spline is used to estimate the locations of the missing landmarks in the target specimen (Gunz et al. 2009).

The second approach (method="Reg") is multivariate regression. Here each landmark with missing values is regressed on all other landmarks for the set of complete specimens, and the missing landmark values are then predicted by this linear regression model. Because the number of variables can exceed the number of specimens, the regression is implemented on scores along the first set of PLS axes for the complete and incomplete blocks of landmarks (see Gunz et al. 2009).

One can also exploit bilateral symmetry to estimate the locations of missing landmarks. Several possibilities exist for implementing this approach (see Gunz et al. 2009). Example R code for one implementation is found in Claude (2008).

NOTE: Because all geometric morphometric analyses and plotting functions implemented in geomorph require a full complement of landmark coordinates, the alternative to estimating the missing landmark coordinates is to proceed with subsequent analyses EXCLUDING specimens with missing values. To do this, see functions `complete.cases` (use: mydata[complete.cases(mydata),]) or `na.omit` (use: newdata <- na.omit(mydata)) to make a dataset of only the complete specimens. These functions require the dataset to be a matrix in the form of a 2d array (see `two.d.array`).

Value

Function returns a n x p matrix of coordinates for the target specimen that includes the original landmarks plus the estimated coordinates for the missing landmarks.

Author(s)

Dean Adams

References

Claude, J. 2008. Morphometrics with R. Springer, New York.

Bookstein, F. L., K. Schafer, H. Prossinger, H. Seidler, M. Fieder, G. Stringer, G. W. Weber, J.-L. Arsuaga, D. E. Slice, F. J. Rohlf, W. Recheis, A. J. Mariam, and L. F. Marcus. 1999. Comparing frontal cranial profiles in archaic and modern Homo by morphometric analysis. Anat. Rec. (New Anat.) 257:217-224.

Gunz, P., P. Mitteroecker, S. Neubauer, G. W. Weber, and F. L. Bookstein. 2009. Principles for the virtual reconstruction of hominin crania. J. Hum. Evol. 57:48-62.

Examples

```
data(plethodon)
plethland<-plethodon$land
  plethland[3,,2]<-plethland[8,,2]<-NA  #create missing landmarks
  plethland[3,,5]<-plethland[8,,5]<-plethland[9,,5]<-NA
  plethland[3,,10]<-NA

estimate.missing(plethland,method="TPS")
estimate.missing(plethland,method="Reg")
```

`findMeanSpec` *Identify specimen closest to the mean of a set of aligned specimens*

Description

A function to identify which specimen lies closest to the estimated mean shape for a set of aligned specimens.

Usage

```
findMeanSpec(A)
```

Arguments

`A`	A 3D array (p x k x n) containing landmark coordinates for a set of aligned specimens

Details

Function takes an array of aligned specimens (such as made by `gpagen`, calculates the distance of each to the estimated mean shape, and returns the name and address of the closest specimen. This function can be used to identify the specimen to be used by `warpRefMesh`.

Value

Function returns the name and address of the specimen closest to the mean of the set of aligned specimens.

Author(s)

Emma Sherratt

See Also

warpRefMesh

`fixed.angle` *Rotate a subset of 2D landmarks to common articulation angle*

Description

A function for rotating a subset of landmarks so that the articulation angle between subsets is constant

Usage

```
fixed.angle(A, art.pt = NULL, angle.pts = NULL, rot.pts = NULL,
  angle = 0, degrees = FALSE)
```

Arguments

`A`	An array (p x k x n) containing landmark coordinates for a set of specimens
`art.pt`	A number specifying which landmark is the articulation point between the two landmark subsets
`angle.pts`	A vector containing numbers specifying which two points used to define the angle (one per subset)
`rot.pts`	A vector containing numbers specifying which landmarks are in the subset to be rotated
`angle`	An optional value specifying the additional amount by which the rotation should be augmented (in radians)
`degrees`	A logical value specifying whether the additional rotation angle is expressed in degrees or radians (radians is default)

Details

This function standardizes the angle between two subsets of landmarks for a set of specimens. The approach assumes a simple hinge-point articulation between the two subsets, and rotates all specimens such that the angle between landmark subsets is equal across specimens (see Adams 1999). As a default, the mean angle is used, though the user may specify an additional amount by which this may be augmented.

Presently, the function is only implemented for two-dimensional landmark data.

Value

Function returns a (p x k x n) array of landmark coordinates.

Author(s)

Dean Adams

References

Adams, D. C. 1999. Methods for shape analysis of landmark data from articulated structures. Evolutionary Ecology Research. 1:959-970.

Examples

```
#Example using Plethodon
#Articulation point is landmark 1, rotate mandibular landmarks (2-5) relative to cranium

data(plethspecies)
fixed.angle(plethspecies$land,art.pt=1,angle.pts=c(5,6),rot.pts=c(2,3,4,5))
```

---

gpagen *Generalized Procrustes analysis of points, curves, and surfaces*

Description

A general function to perform Procrustes analysis of two- or three-dimensional landmark data that can include both fixed landmarks and sliding semilandmarks

Usage

```
gpagen(A, Proj = TRUE, ProcD = TRUE, PrinAxes = TRUE, ShowPlot = TRUE,
  curves = NULL, surfaces = NULL, pointscale = 1)
```

Arguments

A	An array (p x k x n) containing landmark coordinates for a set of specimens
Proj	A logical value indicating whether or not the aligned Procrustes residuals should be projected into tangent space
ProcD	A logical value indicating whether or not Procrustes distance should be used as the criterion for optimizing the positions of semilandmarks
PrinAxes	A logical value indicating whether or not to align the shape data by principal axes
curves	An optional matrix defining which landmarks should be treated as semilandmarks on boundary curves, and which landmarks specify the tangent directions for their sliding
pointscale	An optional value defining the size of the points for all specimens

surfaces — An optional vector defining which landmarks should be treated as semilandmarks on surfaces

ShowPlot — A logical value indicating whether or not a plot of Procrustes residuals should be displayed

Details

The function performs a Generalized Procrustes Analysis (GPA) on two-dimensional or three-dimensional landmark coordinates. The analysis can be performed on fixed landmark points, semilandmarks on curves, semilandmarks on surfaces, or any combination. To include semilandmarks on curves, one must specify a matrix defining which landmarks are to be treated as semilandmarks using the "curves=" option. Likewise, to include semilandmarks on surfaces, one must specify a vector listing which landmarks are to be treated as surface semilandmarks using the "surfaces=" option. The "ProcD=TRUE" option will slide the semilandmarks along their tangent directions using the Procrustes distance criterion, while "ProcD=FALSE" will slide the semilandmarks based on minimizing bending energy. The aligned Procrustes residuals can be projected into tangent space using the "Proj=TRUE" option. NOTE: Large datasets may exceed the memory limitations of R.

Generalized Procrustes Analysis (GPA: Gower 1975, Rohlf and Slice 1990) is the primary means by which shape variables are obtained from landmark data (for a general overview of geometric morphometrics see Bookstein 1991, Rohlf and Marcus 1993, Adams et al. 2004, Zelditch et al. 2012, Mitteroecker and Gunz 2009, Adams et al. 2013). GPA translates all specimens to the origin, scales them to unit-centroid size, and optimally rotates them (using a least-squares criterion) until the coordinates of corresponding points align as closely as possible. The resulting aligned Procrustes coordinates represent the shape of each specimen, and are found in a curved space related to Kendall's shape space (Kendall 1984). Typically, these are projected into a linear tangent space yielding Kendall's tangent space coordinates (Dryden and Mardia 1993, Rohlf 1999), which are used for subsequent multivariate analyses. Additionally, any semilandmarks on curves and are slid along their tangent directions or tangent planes during the superimposition (see Bookstein 1997; Gunz et al. 2005). Presently, two implementations are possible: 1) the locations of semilandmarks can be optimized by minimizing the bending energy between the reference and target specimen (Bookstein 1997), or by minimizing the Procrustes distance between the two (Rohlf 2010).

Value

Function returns a list with the following components:

coords — A (p x k x n) array of aligned Procrustes coordinates, where p is the number of landmark points, k is the number of landmark dimensions (2 or 3), and n is the number of specimens. The third dimension of this array contains names for each specimen if specified in the original input array

Csize — A vector of centroid sizes for each specimen, containing the names for each specimen if specified in the original input array

Author(s)

Dean Adams

References

Adams, D. C., F. J. Rohlf, and D. E. Slice. 2004. Geometric morphometrics: ten years of progress following the 'revolution'. It. J. Zool. 71:5-16.

Adams, D. C., F. J. Rohlf, and D. E. Slice. 2013. A field comes of age: Geometric morphometrics in the 21st century. Hystrix.24:7-14.

Bookstein, F. L. 1991. Morphometric tools for landmark data: Geometry and Biology. Cambridge Univ. Press, New York.

Bookstein, F. L. 1997. Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. 1:225-243.

Dryden, I. L., and K. V. Mardia. 1993. Multivariate shape analysis. Sankhya 55:460-480.

Gower, J. C. 1975. Generalized Procrustes analysis. Psychometrika 40:33-51.

Gunz, P., P. Mitteroecker, and F. L. Bookstein. 2005. semilandmarks in three dimensions. Pp. 73-98 in D. E. Slice, ed. Modern morphometrics in physical anthropology. Klewer Academic/Plenum, New York.

Kendall, D. G. 1984. Shape-manifolds, Procrustean metrics and complex projective spaces. Bulletin of the London Mathematical Society 16:81-121.

Mitteroecker, P., and P. Gunz. 2009. Advances in geometric morphometrics. Evol. Biol. 36:235-247.

Rohlf, F. J., and D. E. Slice. 1990. Extensions of the Procrustes method for the optimal superimposition of landmarks. Syst. Zool. 39:40-59.

Rohlf, F. J., and L. F. Marcus. 1993. A revolution in morphometrics. Trends Ecol. Evol. 8:129-132.

Rohlf, F. J. 1999. Shape statistics: Procrustes superimpositions and tangent spaces. Journal of Classification 16:197-223.

Rohlf, F. J. 2010. tpsRelw: Relative warps analysis. Version 1.49. Department of Ecology and Evolution, State University of New York at Stony Brook, Stony Brook, NY.

Zelditch, M. L., D. L. Swiderski, H. D. Sheets, and W. L. Fink. 2012. Geometric morphometrics for biologists: a primer. 2nd edition. Elsevier/Academic Press, Amsterdam.

Examples

```
#Example 1: fixed points only
data(plethodon)
gpagen(plethodon$land,PrinAxes=FALSE)
points(mshape(gpagen(plethodon$land)$coords),pch=22,col="red",bg="red",cex=1.2)

#Example 2: points and semilandmarks on curves
data(hummingbirds)

#Using Procrustes Distance for sliding
gpagen(hummingbirds$land,curves=hummingbirds$curvepts)

#Using bending energy for sliding
gpagen(hummingbirds$land,curves=hummingbirds$curvepts,ProcD=FALSE)

#Example 3: points, curves and surfaces
```

```
data(scallops)
#Using Procrustes Distance for sliding
gpagen(A=scallops$coorddata, curves=scallops$curvslide, surfaces=scallops$surfslide)
```

`hummingbirds` *Landmark data from hummingbird bills (includes sliding semilandmarks on curves)*

Description

Landmark data from hummingbird bills (includes sliding semilandmarks on curves)

Author(s)

Chelsea Berns and Dean Adams

References

Berns, C.M., and Adams, D.C. 2010. Bill shape and sexual shape dimorphism between two species of temperate hummingbirds: Archilochus alexandri (black-chinned hummingbirds) and Archilochus colubris (ruby-throated hummingbirds). The Auk. 127:626-635.

`morphol.disparity` *Morphological disparity for one or more groups of specimens*

Description

Function estimates morphological disparity and performs pairwise comparisons among groups.

Usage

```
morphol.disparity(A, groups, iter = 999)
```

Arguments

`A`	A matrix (n x [p x k]) or 3D array (p x k x n) containing GPA-aligned coordinates for a set of specimens
`groups`	A factor defining groups
`iter`	Number of iterations for permutation test

Details

The function estimates morphological disparity and performs pairwise comparisons to identify differences between groups. The function takes as input GPA-aligned shape data [e.g., gpagen] and a grouping factor, and estimates disparity as the Procrustes variance for each group, which is the sum of the diagonal elements of the group covariance matrix (e.g., Zelditch et al. 2012). The group Procrustes variances are used as test values, and these are then statistically evaluated through permutation, where the rows of the shape matrix are randomized relative to the grouping variable. The function can be used to obtain disparity for the whole dataset by using a dummy group factor assigning all specimens to one group, in which case only Procrustes variance is returned.

Value

Function returns a list with the following components:

`Disp.obs`	A matrix of Procrustes variance for each group
`Prob.Dist`	A matrix of pairwise significance levels based on permutation

Author(s)

Emma Sherratt

References

Zelditch, M. L., D. L. Swiderski, H. D. Sheets, and W. L. Fink. 2012. Geometric morphometrics for biologists: a primer. 2nd edition. Elsevier/Academic Press, Amsterdam.

Examples

```
data(plethodon)
Y.gpa<-gpagen(plethodon$land)    #GPA-alignment
morphol.disparity(Y.gpa$coords, groups=plethodon$site, iter = 99)
```

`morphol.integr` *Quantify morphological integration between two modules*

Description

Function quantifies the degree of morphological integration between two modules of Procrustes-aligned coordinates

Usage

```
morphol.integr(A1, A2, method = c("PLS", "RV"), warpgrids = TRUE,
  iter = 999, label = NULL, verbose = FALSE)
```

Arguments

A1	A matrix (n x [p1 x k]) or 3D array (p1 x k x n) containing GPA-aligned coordinates for the first module
A2	A matrix (n x [p2 x k]) or 3D array (p2 x k x n) containing GPA-aligned coordinates for the second module
method	Method to estimate morphological integration; see below for details
warpgrids	A logical value indicating whether deformation grids for shapes along PLS1 should be displayed (only relevant if data for A1 or A2 [or both] were input as 3D array)
iter	Number of iterations for significance testing
label	An optional vector indicating labels for each specimen that are to be displayed
verbose	A logical value indicating whether the output is basic or verbose (method="PLS" only) (see Value below)

Details

The function quantifies the degree of morphological integration between two modules of shape data as defined by landmark coordinates. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with gpagen]. The function may be used to assess the degree of morphological integration between two separate structures or between two modules defined within the same landmark configuration.

Two analytical approaches are currently implemented to assess the degree of morphological integration:

1. method="PLS" (default) the function estimates the degree of morphological integration using two-block partial least squares, or PLS. When used with landmark data, this analysis is referred to as singular warps analysis (Bookstein et al. 2003). When method="PLS", the scores along the X & Y PLS axes are also returned, as is a plot of PLS scores from Block1 versus Block2 along the first set of PLS axes. Thin-plate spline deformation grids along these axes are also shown (if data were input as a 3D array). Note: deformation grids are displayed for each block of landmarks separately. If the two blocks of landmarks are derived from a single structure (i.e. a single landmark configuration), one can plot the overall deformation along PLS1 using the procedure provided in the example below.
2. method="RV" the function estimates the degree of morphological integration using the RV coefficient (Klingenberg 2009). Significance testing for both approaches is found by permuting the objects in one data matrix relative to those in the other. A histogram of coefficients obtained via resampling is presented, with the observed value designated by an arrow in the plot.

If evaluating an a priori hypothesis of modularity within a structure is of interest, one may use the average RV coefficient as implemented in the function compare.modular.partitions.

Value

Function returns the a list with the following components:

value	The estimate of morphological integration: PLS.corr or RV

pvalue	The significance level of the observed signal
Xscores	PLS scores for the first block of landmarks (method="PLS" only when verbose=TRUE)
Yscores	PLS scores for the second block of landmarks (method="PLS" only when verbose=TRUE)

Author(s)

Dean Adams

References

Bookstein, F. L., P. Gunz, P. Mitteroecker, H. Prossinger, K. Schaefer, and H. Seidler. 2003. Cranial integration in Homo: singular warps analysis of the midsagittal plane in ontogeny and evolution. J. Hum. Evol. 44:167-187.

Klingenberg, C. P. 2009. Morphometric integration and modularity in configurations of landmarks: tools for evaluating a priori hypotheses. Evol. Develop. 11:405-421.

Examples

```
data(plethodon)
Y.gpa<-gpagen(plethodon$land)    #GPA-alignment

#Morphological integration using PLS between two modules within a structure
morphol.integr(Y.gpa$coords[1:5,,],Y.gpa$coords[6:12,,],method="PLS",iter=99)

#Morphological integration using RV between two modules within a structure
morphol.integr(Y.gpa$coords[1:5,,],Y.gpa$coords[6:12,,],method="RV",iter=99)

#Deformation plot for case when both blocks are derived from the same landmark configuration
res<-morphol.integr(Y.gpa$coords[1:5,,],Y.gpa$coords[6:12,,],method="PLS",iter=99,verbose=TRUE)
  ref<-mshape(Y.gpa$coords)   #overall reference
 plotRefToTarget(ref,Y.gpa$coords[,,which.min(res$x.scores)],method="TPS") #Min along PLS1
 plotRefToTarget(ref,Y.gpa$coords[,,which.max(res$x.scores)],method="TPS") #Max along PLS1
```

---

mosquito *landmarks on mosquito wings*

Description

landmarks on mosquito wings

Author(s)

Dean Adams

motionpaths *Simulated motion paths*

Description

Simulated motion paths

Author(s)

Dean Adams

References

Adams, D. C., and M. L. Collyer. 2009. A general framework for the analysis of phenotypic trajectories in evolutionary studies. Evolution 63:1143-1154.

mshape *Estimate mean shape for a set of aligned specimens*

Description

Estimate the mean shape for a set of aligned specimens

Usage

```
mshape(A)
```

Arguments

A	An array (p x k x n) containing GPA-aligned coordinates for a set of specimens

Details

The function estimates the average landmark coordinates for a set of aligned specimens. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with gpagen]. This function is described in Claude (2008).

Author(s)

Julien Claude

References

Claude, J. 2008. Morphometrics with R. Springer, New York.

Examples

```
data(plethodon)
Y.gpa<-gpagen(plethodon$land)    #GPA-alignment

mshape(Y.gpa$coords)   #mean (consensus) configuration
```

`pairwise.slope.test` *Pairwise Comparisons of Slopes*

Description

Function performs pairwise comparisons among slopes for groups as specified by a linear model.

Usage

```
pairwise.slope.test(f1, iter = 999, het.slopes = T, angle.type = "r",
  RRPP = FALSE)
```

Arguments

`f1`	A formula for the linear model from which groups are to be compared (e.g., y~x1+x2)
`iter`	Number of iterations for permutation test
`het.slopes`	A logical value indicting whether slopes are to be compared
`angle.type`	A value specifying whether differences between slopes should be represented by vector correlations (r), radians (rad) or degrees (deg)
`RRPP`	a logical value indicating whether residual randomization should be used for significance testing

Details

The function performs pairwise comparisons to identify differences in slopes between groups. The function is designed as a post-hoc test to MANCOVA, where the latter has identified significant shape variation explained by a covariate*group interaction term.

As input the user provides a formula describing the linear model of how shape varies as a function of several explanatory variables. This MUST be in the form of: [y~covariate + group], and the shape data (y) must be in the form of a two-dimensional data matrix of dimension (n x [p x k]), rather than a 3D array. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with `gpagen`]. The function `two.d.array` can be used to obtain a two-dimensional data matrix from a 3D array of landmark coordinates. From the data, the slopes for each group are estimated, and pairwise differences in slopes determined.

For the model, one can specify whether slopes or intercepts are to be evaluated. Slopes are compared if (heterogenous slopes) het.slopes=TRUE.To evaluate significance of the pairwise differences, two possible resampling procedures are provided. First, if RRPP=FALSE, the rows of the matrix of

shape variables are randomized relative to the design matrix. This is analogous to a 'full' randomization. Second, if RRPP=TRUE, a residual randomization permutation procedure is utilized (Collyer et al. 2014). Here, residual shape values from a reduced model are obtained, and are randomized with respect to the linear model under consideration. These are then added to predicted values from the remaining effects to obtain pseudo-values from which SS are calculated. NOTE: for single-factor designs, the two approaches are identical. However, when evaluating factorial models it has been shown that RRPP attains higher statistical power and thus has greater ability to identify patterns in data should they be present (see Anderson and terBraak 2003).

Value

Function returns a list with the following components:

`ANOVA.table`	An ANOVA table assessing the linear model
`Obs.LS.Dist`	A matrix of pairwise differences between intercepts (least squares means) if het.slopes = FALSE
`Slope.Dist`	A matrix of pairwise differences between slopes represented as correlations, radians, or degrees, if het.slopes = TRUE
`Prob.Dist`	A matrix of pairwise significance levels based on permutation for either intercepts or slopes
`Magnitude.Diff`	A matrix of pairwise differences in magnitude between regression lines (from smallest to largest specimen, if het.slopes = TRUE)
`Prob.Mag`	A matrix of pairwise significance levels based on permutation for magnitude differences

Author(s)

Mike Collyer

References

Anderson MJ. and C.J.F. terBraak. 2003. Permutation tests for multi-factorial analysis of variance. Journal of Statistical Copmutation and Simulation 73: 85-113.

Collyer, M.L., D.J. Sekora, and D.C. Adams. 2014. A method for analysis of phenotypic change for phenotypes described by high-dimensional data. Heredity. (In Press).

Examples

```
### MANCOVA example for Goodall's F test (multivariate shape vs. factors)
data(plethodon)
Y.gpa<-gpagen(plethodon$land)    #GPA-alignment
y<-two.d.array(Y.gpa$coords)

## Pairwise slope test
# Assuming heterogenous slopes
pairwise.slope.test(y~Y.gpa$Csize+plethodon$site,iter=49,angle.type="rad")

# Assuming parallel slopes
pairwise.slope.test(y~Y.gpa$Csize+plethodon$site,het.slopes=FALSE, iter=49, angle.type="rad")
```

```
## Using RRPP
# Assuming heterogenous slopes
pairwise.slope.test(y~Y.gpa$Csize+plethodon$site,iter=49, angle.type="rad", RRPP=TRUE)
# Assuming parallel slopes
pairwise.slope.test(y~Y.gpa$Csize+plethodon$site, het.slopes=FALSE,
      iter=49, angle.type="rad", RRPP=TRUE)
```

`pairwiseD.test` *Pairwise Group Comparisons*

Description

Function performs pairwise comparisons among groups using the Euclidean distances among group means.

Usage

```
pairwiseD.test(f1, iter = 999, RRPP = FALSE)
```

Arguments

`f1`	A formula for the linear model from which groups are to be compared (e.g., y~x1+x2)
`iter`	Number of iterations for permutation test
`RRPP`	a logical value indicating whether residual randomization should be used for significance testing

Details

The function performs pairwise comparisons to identify shape differences among groups. The function is designed as a post-hoc test to Procrustes ANOVA, where the latter has identified significant shape variation explained by a grouping factor.

As input the user provides a formula describing the linear model of how shape (y) varies as a function of groups (x). Multiple factors may be examined. The shape data (y) must be in the form of a two-dimensional data matrix of dimension (n x [p x k]), rather than a 3D array. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with `gpagen`]. The function `two.d.array` can be used to obtain a two-dimensional data matrix from a 3D array of landmark coordinates. From the data, the Euclidean distances among group means are estimated, and used as test values.

To evaluate significance of group differences, two possible resampling procedures are provided. First, if RRPP=FALSE, the rows of the matrix of shape variables are randomized relative to the design matrix. This is analogous to a 'full' randomization. Second, if RRPP=TRUE, a residual randomization permutation procedure is utilized (Collyer et al. 2014). Here, residual shape values from a reduced model are obtained, and are randomized with respect to the linear model under consideration. These are then added to predicted values from the remaining effects to obtain pseudo-values from which SS are calculated. NOTE: for single-factor designs, the two approaches are

identical. However, when evaluating factorial models it has been shown that RRPP attains higher statistical power and thus has greater ability to identify patterns in data should they be present (see Anderson and terBraak 2003).

Value

Function returns a list with the following components:

`Dist.obs`	A matrix of Euclidean distances among group means
`Prob.Dist`	A matrix of pairwise significance levels based on permutation

Author(s)

Mike Collyer and Dean Adams

References

Anderson MJ. and C.J.F. terBraak. 2003. Permutation tests for multi-factorial analysis of variance. Journal of Statistical Copmutation and Simulation 73: 85-113.

Collyer, M.L., D.J. Sekora, and D.C. Adams. 2014. A method for analysis of phenotypic change for phenotypes described by high-dimensional data. Heredity. (In Press).

Examples

```
data(plethodon)
Y.gpa<-gpagen(plethodon$land)    #GPA-alignment
y<-two.d.array(Y.gpa$coords)
### Procrustes ANOVA
procD.lm(y~plethodon$species,iter=99)

### Pairwise comparisons: full randomization
pairwiseD.test(y~plethodon$species*plethodon$site,iter=99)

## Pairwise comparisons: residual randomization
#' pairwiseD.test(y~plethodon$species*plethodon$site,iter=99,RRPP=TRUE)
```

---

phylo.pls *Quantify phylogenetic morphological integration between two sets of variables*

Description

Function quantifies the degree of phylogenetic morphological covariation between two sets of Procrustes-aligned coordinates using partial least squares.

Usage

```
phylo.pls(A1, A2, phy, warpgrids = TRUE, iter = 999, label = NULL,
  verbose = FALSE)
```

Arguments

A1	A 2D array (n x [p1 x k1]) or 3D array (p1 x k1 x n) containing landmark coordinates for the first block
A2	A 2D array (n x [p2 x k2]) or 3D array (p2 x k2 x n) containing landmark coordinates for the second block
phy	A phylogenetic tree of class phylo - see read.tree in library ape
warpgrids	A logical value indicating whether deformation grids for shapes along PC1 should be displayed (only relevant if data for A1 or A2 [or both] were input as 3D array)
iter	Number of iterations for significance testing
label	An optional vector indicating labels for each specimen that are to be displayed
verbose	A logical value indicating whether the output is basic or verbose (see Value below)

Details

The function quantifies the degree of phylogenetic morphological integration between two sets of shape data as defined by landmark coordinates. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with gpagen].

The function estimates the degree of morphological covariation between two sets of variables while accounting for phylogeny using partial least squares (Adams and Felice 2014). The observed value is statistically assessed using permutation, where data for one block are permuted across the tips of the phylogeny, an estimate of the covariation between sets of variables, and compared to the observed value.

A plot of PLS scores from Block1 versus Block2 is provided for the first set of PLS axes. Thin-plate spline deformation grids along these axes are also shown (if data were input as a 3D array).

Value

Function returns a list with the following components:

PLS Correlation	The estimate of phylogenetic morphological covariation
pvalue	The significance level of the observed signal
Block 1 PLS Scores	PLS scores for the first block of landmarks (when verbose=TRUE)
Block 2 PLS Scores	PLS scores for the second block of landmarks (when verbose=TRUE)

Author(s)

Dean Adams

References

Adams, D.C. and R. Felice. 2014. Assessing phylogenetic morphological integration and trait covariation in morphometric data using evolutionary covariance matrices. PLOS ONE. 9(4):e94335.

Examples

```
data(plethspecies)
Y.gpa<-gpagen(plethspecies$land)    #GPA-alignment

phylo.pls(Y.gpa$coords[1:5,,],Y.gpa$coords[6:11,,],plethspecies$phy,iter=5)
```

`physignal` *Assessing phylogenetic signal in morphometric data*

Description

Function calculates the degree of phylogenetic signal from a set of Procrustes-aligned specimens

Usage

```
physignal(phy, A, iter = 249, method = c("Kmult", "SSC"))
```

Arguments

`phy`	A phylogenetic tree of class phylo - see `read.tree` in library ape
`A`	A matrix (n x [p x k]) or 3D array (p x k x n) containing GPA-aligned coordinates for a set of specimens
`iter`	Number of iterations for significance testing
`method`	Method for estimating phylogenetic signal (Kmult or SSC)

Details

The function estimates the degree of phylogenetic signal present in shape data for a given phylogeny. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with `gpagen`]. Two approaches may be used to quantify phylogenetic signal. First, a multivariate version of the K-statistic may be utilized (Kmult: Adams 2014). This value evaluates the degree of phylogenetic signal in a dataset relative to what is expected under a Brownian motion model of evolution. For geometric morphometric data, the approach is a mathematical generalization of the Kappa statistic (Blomberg et al. 2003) appropriate for highly multivariate data (see Adams 2014).

The second approach estimates phylogenetic signal as the sum of squared changes (SSC) in shape along all branches of the phylogeny (Klingenberg and Gidaszewski 2010). Significance testing is found by permuting the shape data among the tips of the phylogeny. Note that this method can be quite slow as ancestral states must be estimated for every iteration.

For both approaches a plot of the specimens in tangent space with the phylogeny superimposed is included (NOTE: if ancestral states are desired, run `plotGMPhyloMorphoSpace`).

The tree must have number of tips equal to number of taxa in the data matrix (e.g. `drop.tip`). And, tip labels of the tree MUST be exactly the same as the taxa names in the landmark data matrix (check using `match`).

This function can be used with univariate data (i.e. centroid size) if imported as matrix with rownames giving the taxa names.

Value

Function returns a list with the following components:

phy.signal	The estimate of phylogenetic signal
pvalue	The significance level of the observed signal

Author(s)

Dean Adams

References

Blomberg SP, Garland T, Ives AR. 2003. Testing for phylogenetic signal in comparative data: behavioral traits are more labile. Evolution, 57:717-745.

Klingenberg, C. P., and N. A. Gidaszewski. 2010. Testing and quantifying phylogenetic signals and homoplasy in morphometric data. Syst. Biol. 59:245-261.

Adams, D.C. 2014. A generalized K statistic for estimating phylogenetic signal from shape and other high-dimensional multivariate data. Systematic Biology. 63. DOI:10.1093/sysbio/syu030.

Examples

```
data(plethspecies)
Y.gpa<-gpagen(plethspecies$land)    #GPA-alignment

#Test for phylogenetic signal in shape
physignal(plethspecies$phy,Y.gpa$coords,method="Kmult",iter=99)

#Test for phylogenetic signal in size
Csize <- matrix(Y.gpa$Csize, dimnames=list(names(Y.gpa$Csize))) # make matrix Csize with names
physignal(plethspecies$phy,Csize,method="Kmult",iter=99)
```

---

plethodon *Landmark data from Plethodon salamander heads*

Description

Landmark data from Plethodon salamander heads

Author(s)

Dean Adams

References

Adams, D. C. 2004. Character displacement via aggressive interference in Appalachian salamanders. Ecology. 85:2664-2670.

Adams, D.C. 2010. Parallel evolution of character displacement driven by competitive selection in terrestrial salamanders. BMC Evolutionary Biology. 10(72)1-10.

plethShapeFood — *Head shape and food use data from Plethodon salamanders*

Description

Head shape and food use data from Plethodon salamanders

Author(s)

Dean Adams

References

Adams, D. C., and F. J. Rohlf. 2000. Ecological character displacement in Plethodon: biomechanical differences found from a geometric morphometric study. Proceedings of the National Academy of Sciences, U.S.A. 97:4106-4111

plethspecies — *Average head shape and phylogenetic relationships for several Plethodon salamander species*

Description

Average head shape and phylogenetic relationships for several Plethodon salamander species

Author(s)

Dean Adams

References

Phylogeny pruned from: Wiens et al. (2006). Evol.

Data from: Adams and Rohlf (2000); Adams et al. (2007); Arif et al. (2007) Myers and Adams (2008)

`plotAllometry` *Plot allometric patterns in landmark data*

Description

Function plots allometry curves for a set of specimens

Usage

```
plotAllometry(A, sz, groups = NULL, method = c("CAC", "RegScore",
  "PredLine"), warpgrids = TRUE, iter = 249, label = NULL, mesh = NULL,
  logsz = TRUE, verbose = FALSE)
```

Arguments

`A`	An array (p x k x n) containing landmark coordinates for a set of specimens
`sz`	A vector of centroid size measures for all specimens
`groups`	An optional vector containing group labels for each specimen if available
`method`	Method for estimating allometric shape components; see below for details
`warpgrids`	A logical value indicating whether deformation grids for small and large shapes should be displayed (note: if groups are provided no TPS grids are shown)
`iter`	Number of iterations for significance testing
`label`	An optional vector indicating labels for each specimen that are to be displayed
`mesh`	A mesh3d object to be warped to represent shape deformation of the directional and fluctuating components of asymmetry if warpgrids= TRUE (see `warpRefMesh`).
`logsz`	A logical value indicating whether the log(centroid size) is used
`verbose`	A logical value indicating whether the output is basic or verbose (see Value below)

Details

The function performs a regression of shape on size, and generates a plot that describes the multivariate relationship between size and shape derived from landmark data (i.e., allometry). It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with `gpagen`]. The abscissa of the plot is log(centroid size) while the ordinate represents shape [NOTE: the function takes the input size and performed log-transformation automatically by default (logsz = TRUE), as log(CSize) should be used]. Three complementary approaches can be implemented to visualize allometry:

1. If "method=CAC" (the default) the function calculates the common allometric component of the shape data, which is an estimate of the average allometric trend within groups (Mitteroecker et al. 2004). The function also calculates the residual shape component (RSC) for the data.

2. If "method=RegScore" the function calculates shape scores from the regression of shape on size, and plots these versus size (Drake and Klingenberg 2008). For a single group, these shape scores are mathematically identical to the CAC (Adams et al. 2013).
3. If "method=PredLine" the function calculates predicted values from a regression of shape on size, and plots the first principal component of the predicted values versus size as a stylized graphic of the allometric trend (Adams and Nistri 2010).

For all methods, both centroid size and allometry scores are returned. Optionally, deformation grids can be requested, which display the shape of the smallest and largest specimens relative to the average specimen (using 'warpgrids=T' or 'warpgrids=F'). Finally, if groups are provided, the above approaches are implemented while accounting for within-group patterns of covariation (see references for explanation). In this case, the regression is of the form: shape~size+groups (Note: to examine the interaction term use `procD.lm`). Specimens from each group are plotted using distinct colors based on the order in which the groups are found in the dataset, and using R's standard color palette: black, red, green, blue, cyan, magenta, yellow, and gray. NOTE: to change the colors of the groups, simply substitute a vector of the desired colors for each specimen.

Value

Function returns an ANOVA table of statistical results for log centroid size: df, SS, MS, Prand. If verbose=TRUE, function returns a list with the following components:

`ProcDist.lm`	An ANOVA table as above
`allom.score`	A matrix of the allometry shape scores
`logCsize`	A matrix of log centroid size
`pred.shape`	A matrix containing the predicted shapes from the regression
`resid.shape`	The residual shape component (RSC) of the data ("method=CAC" only)

Author(s)

Dean Adams

References

Adams, D.C., F.J. Rohlf, and D.E. Slice. 2013. A field comes of age: geometric morphometrics in the 21st century. Hystrix. 24:7-14.

Adams, D. C., and A. Nistri. 2010. Ontogenetic convergence and evolution of foot morphology in European cave salamanders (Family: Plethodontidae). BMC Evol. Biol. 10:1-10.

Drake, A. G., and C. P. Klingenberg. 2008. The pace of morphological change: Historical transformation of skull shape in St Bernard dogs. Proceedings of the Royal Society B, Biological Sciences 275:71'76.

Mitteroecker, P., P. Gunz, M. Bernhard, K. Schaefer, and F. L. Bookstein. 2004. Comparison of cranial ontogenetic trajectories among great apes and humans. J. Hum. Evol. 46:679-698.

Examples

```
data(ratland)
Y.gpa<-gpagen(ratland)    #GPA-alignment

#Using CAC for plot
plotAllometry(Y.gpa$coords,Y.gpa$Csize,method="CAC", iter=5)

#Using Regression Scores for plot
plotAllometry(Y.gpa$coords,Y.gpa$Csize,method="RegScore", iter=5)

#Using predicted allometry curve for plot
plotAllometry(Y.gpa$coords,Y.gpa$Csize,method="PredLine", iter=5)
```

---

plotAllSpecimens *Plot landmark coordinates for all specimens*

Description

Function plots landmark coordinates for a set of specimens

Usage

```
plotAllSpecimens(A, mean = TRUE, links = NULL, pointscale = 1,
  meansize = 2)
```

Arguments

A	An array (p x k x n) containing GPA-aligned coordinates for a set of specimens
mean	A logical value indicating whether the mean shape should be included in the plot
links	An optional matrix defining for links between landmarks
pointscale	An optional value defining the size of the points for all specimens
meansize	An optional value defining the size of the points representing the average specimen

Details

The function creates a plot of the landmark coordinates for all specimens. This is useful for examining patterns of shape variation after GPA. If "mean=TRUE", the mean shape will be calculated and added to the plot. Additionally, if a matrix of links is provided, the landmarks of the mean shape will be connected by lines. The link matrix is an m x 2 matrix, where m is the desired number of links. Each row of the link matrix designates the two landmarks to be connected by that link. The function will plot either two- or three-dimensional data.

Author(s)

Dean Adams

Examples

```
data(plethodon)
Y.gpa<-gpagen(plethodon$land)    #GPA-alignment

plotAllSpecimens(Y.gpa$coords,links=plethodon$links)
```

`plotGMPhyloMorphoSpace` *Plot phylogenetic tree and specimens in tangent space*

Description

Function plots a phylogenetic tree and a set of Procrustes-aligned specimens in tangent space

Usage

```
plotGMPhyloMorphoSpace(phy, A, labels = TRUE, ancStates = TRUE)
```

Arguments

`phy`	A phylogenetic tree of class phylo - see `read.tree` in library ape
`A`	A matrix (n x [p x k]) or 3D array (p x k x n) containing GPA-aligned coordinates for a set of specimens
`labels`	A logical value indicating whether taxa labels (tips and ancestors) should be included
`ancStates`	A logical value indicating whether ancestral state values should be returned

Details

The function creates a plot of the first two dimensions of tangent space for a set of Procrustes-aligned specimens. The phylogenetic tree for these specimens is superimposed in this plot revealing how shape evolves (e.g., Rohlf 2002; Klingenberg and Gidaszewski 2010). The plot also displays the ancestral states for each node of the phylogenetic tree (obtained from `fastAnc`), whose values can optionally be returned.

Value

Function returns estimated ancestral states if ancStates=TRUE

Author(s)

Dean Adams & Emma Sherratt

References

Klingenberg, C. P., and N. A. Gidaszewski. 2010. Testing and quantifying phylogenetic signals and homoplasy in morphometric data. Syst. Biol. 59:245-261.

Rohlf, F. J. 2002. Geometric morphometrics and phylogeny. Pp. 175'193 in N. Macleod, and P. Forey, eds. Morphology, shape, and phylogeny. Taylor & Francis, London.

Examples

```
data(plethspecies)
Y.gpa<-gpagen(plethspecies$land)    #GPA-alignment

plotGMPhyloMorphoSpace(plethspecies$phy,Y.gpa$coords)
```

`plotRefToTarget` *Plot shape differences between a reference and target specimen*

Description

Function plots shape differences between a reference and target specimen

Usage

```
plotRefToTarget(M1, M2, mesh = NULL, method = c("TPS", "vector", "points",
  "surface"), mag = 1, links = NULL, ...)
```

Arguments

`M1`	Matrix of landmark coordinates for the first (reference) specimen
`M2`	Matrix of landmark coordinates for the second (target) specimen
`mesh`	A mesh3d object for use with method="surface"
`method`	Method used to visualize shape difference; see below for details
`mag`	The desired magnification to be used when visualizing the shape difference (e.g., mag=2)
`links`	An optional matrix defining for links between landmarks
`...`	Additional parameters to be passed to `plot`, `plot3d` or `shade3d`.

Details

The function generates a plot of the shape differences of a target specimen relative to a reference specimen. The option mag allows the user to indicates the degree of magnification to be used when displaying the shape difference. The function will plot either two- or three-dimensional data. Four methods for plots are available:

1. TPS a thin-plate spline deformation grid is generated. For 3D data, this method will generate thin-plate spline deformations in the x-y and x-z planes.

2. vector: a plot showing the vector displacements between corresponding landmarks in the reference and target specimen is shown.
3. points a plot is displayed with the landmarks in the target (black) overlaying those of the reference (gray). Additionally, if a matrix of links is provided, the landmarks of the mean shape will be connected by lines. The link matrix is an M x 2 matrix, where M is the desired number of links. Each row of the link matrix designates the two landmarks to be connected by that link.
4. surface a mesh3d surface is warped using thin-plate spline (for 3D data only). Requires mesh3d object in option mesh, made using warpRefMesh.

This function combines numerous plotting functions found in Claude (2008).

Value

If using method="surface", function will return the warped mesh3d object.

Author(s)

Dean Adams & Emma Sherratt

References

Claude, J. 2008. Morphometrics with R. Springer, New York.

Examples

```
data(plethodon)
Y.gpa<-gpagen(plethodon$land)    #GPA-alignment
ref<-mshape(Y.gpa$coords)

# Different plotting options
plotRefToTarget(ref,Y.gpa$coords[,,39])

plotRefToTarget(ref,Y.gpa$coords[,,39],mag=3)   #magnify difference by 3X

plotRefToTarget(ref,Y.gpa$coords[,,39],method="vector")

plotRefToTarget(ref,Y.gpa$coords[,,39],method="points")

# Three dimensional data

data(scallops)
Y.gpa<-gpagen(A=scallops$coorddata, curves=scallops$curvslide, surfaces=scallops$surfslide)
ref<-mshape(Y.gpa$coords)
plotRefToTarget(ref,Y.gpa$coords[,,1],method="points")
```

plotspec *Plot 3D specimen, fixed landmarks and surface semilandmarks.*

Description

A function to plot three-dimensional (3D) specimen along with its landmarks.

Usage

```
plotspec(spec, digitspec, fixed = NULL, ptsize = 1, centered = FALSE, ...)
```

Arguments

spec	An object of class shape3d/mesh3d, or matrix of 3D vertex coordinates.
digitspec	Name of data matrix containing 3D fixed and/or surface sliding coordinates.
fixed	Numeric The number of fixed template landmarks (listed first in digitspec)
ptsize	Numeric Size to plot the mesh points (vertices), e.g. 0.1 for dense meshes, 3 for sparse meshes
centered	Logical Whether the data matrix is in the surface mesh coordinate system (centered=FALSE) or if the data were collected after the mesh was centered (centered=TRUE)- see details.
...	additional parameters which will be passed to plot3d.

Details

Function to plot 3D specimens along with their digitized "fixed" landmarks and semilandmarks "surface sliders" and "curve sliders". If specimen is a 3D surface (class shape3d/mesh3d) mesh is plotted. For visualization purposes, 3D coordinate data collected using digit.fixed or digitsurface and buildtemplate prior to build 1.1-6 were centered by default. Therefore use this function with centered=TRUE. Data collected outside geomorph should be read using centered=FALSE. The function assumes the fixed landmarks are listed at the beginning of the coordinate matrix (digitspec).

Author(s)

Erik Otarola-Castillo & Emma Sherratt

See Also

warpRefMesh

read.ply

Examples

```
data(scallopPLY)
ply <- scallopPLY$ply
digitdat <- scallopPLY$coords
plotspec(spec=ply,digitspec=digitdat,fixed=16, centered =TRUE)
```

`plotTangentSpace` *Plot specimens in tangent space*

Description

Function plots a set of Procrustes-aligned specimens in tangent space along their principal axes

Usage

```
plotTangentSpace(A, axis1 = 1, axis2 = 2, warpgrids = TRUE, mesh = NULL,
  label = NULL, groups = NULL, verbose = FALSE)
```

Arguments

`A`	An array (p x k x n) containing landmark coordinates for a set of aligned specimens
`warpgrids`	A logical value indicating whether deformation grids for shapes along X-axis should be displayed
`mesh`	A mesh3d object to be warped to represent shape deformation along X-axis (when warpgrids=TRUE) as described in `plotRefToTarget`.
`axis1`	A value indicating which PC axis should be displayed as the X-axis (default = PC1)
`axis2`	A value indicating which PC axis should be displayed as the Y-axis (default = PC2)
`label`	An optional vector indicating labels for each specimen are to be displayed
`groups`	An optional factor vector specifying group identity for each specimen
`verbose`	A logical value indicating whether the output is basic or verbose (see Value below)

Details

The function performs a principal components analysis of shape variation and plots two dimensions of tangent space for a set of Procrustes-aligned specimens (default is PC1 vs. PC2). The percent variation along each PC-axis is returned. Additionally (and optionally, warpgrids=T), deformation grids can be requested, which display the shape of specimens at the ends of the range of variability along PC1. If groups are provided, specimens from each group are plotted using distinct colors based on the order in which the groups are found in the dataset, and using R's standard color palette: black, red, green, blue, cyan, magenta, yellow, and gray. NOTE: to change the colors of the groups, simply substitute a vector of the desired colors for each specimen (see example below).

Value

Function returns a table summarizing the percent variation explained by each pc axis (equivalent to summary() of prcomp) (default, verbose = FALSE). If verbose=TRUE, function returns a list containing the following components:

pc.summary	A PC summary table as above
pc.scores	The set of principal component scores for all specimens.
pc.shapes	A list with four components of the shape coordinates of the extreme ends of axis1 and axis2 is returned, which can be used by warpRefMesh.

Author(s)

Dean Adams & Emma Sherratt

Examples

```
data(plethodon)
Y.gpa<-gpagen(plethodon$land)    #GPA-alignment

gp <- as.factor(paste(plethodon$species, plethodon$site)) # group must be a factor
plotTangentSpace(Y.gpa$coords, groups = gp)

##To change colors of groups
col.gp<-c(rep("black",10),rep("red",10),rep("yellow",10),rep("orange",10)) # must not be a factor

plotTangentSpace(Y.gpa$coords, groups = col.gp)
```

---

procD.lm *Procrustes ANOVA/regression for shape data*

Description

Function performs Procrustes ANOVA with permutation procedures to assess statistical hypotheses describing patterns of shape variation and covariation for a set of Procrustes-aligned coordinates

Usage

```
procD.lm(f1, iter = 999, RRPP = FALSE, verbose = FALSE)
```

Arguments

f1	A formula for the linear model (e.g., y~x1+x2)
iter	Number of iterations for significance testing
RRPP	a logical value indicating whether residual randomization should be used for significance testing
verbose	A logical value specifying whether additional output should be displayed

Details

The function quantifies the relative amount of shape variation attributable to one or more factors in a linear model and assesses this variation via permutation. Data input is specified by a formula (e.g., y~X), where 'y' specifies the response variables (shape data), and 'X' contains one or more independent variables (discrete or continuous). The response matrix 'y' must be in the form of a two-dimensional data matrix of dimension (n x [p x k]), rather than a 3D array. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with gpagen]. The function two.d.array can be used to obtain a two-dimensional data matrix from a 3D array of landmark coordinates. The names specified for the independent (x) variables in the formula represent one or more vectors containing continuous data or factors. It is assumed that the order of the specimens in the shape matrix matches the order of values in the independent variables.

The function performs statistical assessment of the terms in the model using Procrustes distances among specimens, rather than explained covariance matrices among variables. With this approach, the sum-of-squared Procrustes distances are used as a measure of SS (see Goodall 1991). The observed SS are evaluated through permutation. In morphometrics this approach is known as a Procrustes ANOVA (Goodall 1991), which is equivalent to distance-based anova designs (Anderson 2001). Two possible resampling procedures are provided. First, if RRPP=FALSE, the rows of the matrix of shape variables are randomized relative to the design matrix. This is analogous to a 'full' randomization. Second, if RRPP=TRUE, a residual randomization permutation procedure is utilized (Collyer et al. 2014). Here, residual shape values from a reduced model are obtained, and are randomized with respect to the linear model under consideration. These are then added to predicted values from the remaining effects to obtain pseudo-values from which SS are calculated. NOTE: for single-factor designs, the two approaches are identical. However, when evaluating factorial models it has been shown that RRPP attains higher statistical power and thus has greater ability to identify patterns in data should they be present (see Anderson and terBraak 2003).

Value

Function returns an ANOVA table of statistical results for all factors: df (for each factor), SS, MS, F ratio, Prand, and Rsquare.

Author(s)

Dean Adams and Mike Collyer

References

Anderson MJ. 2001. A new method for non-parametric multivariate analysis of variance. Austral Ecology 26: 32-46.

Anderson MJ. and C.J.F. terBraak. 2003. Permutation tests for multi-factorial analysis of variance. Journal of Statistical Copmutation and Simulation 73: 85-113.

Collyer, M.L., D.J. Sekora, and D.C. Adams. 2014. A method for analysis of phenotypic change for phenotypes described by high-dimensional data. Heredity. (In Press).

Goodall, C. R. 1991. Procrustes methods in the statistical analysis of shape. Journal of the Royal Statistical Society B 53:285-339.

Examples

```
### MANOVA example for Goodall's F test (multivariate shape vs. factors)
data(plethodon)
Y.gpa<-gpagen(plethodon$land)    #GPA-alignment
y<-two.d.array(Y.gpa$coords)

procD.lm(y~plethodon$species*plethodon$site,iter=99)

### Regression example
data(ratland)
rat.gpa<-gpagen(ratland)         #GPA-alignment

procD.lm(two.d.array(rat.gpa$coords)~rat.gpa$Csize,iter=99)

## using RRPP
 procD.lm(two.d.array(rat.gpa$coords)~rat.gpa$Csize,iter=99,RRPP=TRUE)
```

---

procD.pgls *Phylogenetic ANOVA/regression for shape data*

Description

Function performs Procrustes ANOVA in a phylogenetic framework and uses permutation procedures to assess statistical hypotheses describing patterns of shape variation and covariation for a set of Procrustes-aligned coordinates

Usage

```
procD.pgls(f1, phy, iter = 999)
```

Arguments

f1	A formula for the linear model (e.g., y~x1+x2)
phy	A phylogenetic tree of class phylo - see read.tree in library ape
iter	Number of iterations for significance testing

Details

The function performs ANOVA and regression models in a phylogenetic context under a Brownian motion model of evolution, in a manner that can accommodate high-dimensional datasets. The approach is derived from the statistical equivalency between parametric methods utilizing covariance matrices and methods based on distance matrices (Adams 2014). Data input is specified by a formula (e.g., y~X), where 'y' specifies the response variables (shape data), and 'X' contains one or more independent variables (discrete or continuous). The response matrix 'y' must be in the form of a two-dimensional data matrix of dimension (n x [p x k]), rather than a 3D array. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with gpagen]. The user must also specify a phylogeny describing the evolutionary

relationships among species (of class phylo). Note that the specimen labels for both x and y must match the labels on the tips of the phylogeny.

From the phylogeny, a phylogenetic transformation matrix is obtained under a Brownian motion model, and used to transform the x and y variables. Next, the Gower-centered distance matrix is obtained from predicted values from the model (y~x), from which sums-of-squares, F-ratios, and R^2 are estimated for each factor in the model (see Adams, 2014). Data are then permuted across the tips of the phylogeny, and estimates of statistical values are obtained for the permuted data, which are compared to the observed value to assess significance.

Value

Function returns an ANOVA table of statistical results for all factors: df (for each factor), SS, MS, F ratio, Prand, and Rsquare.

Author(s)

Dean Adams

References

Adams, D.C. 2014. A method for assessing phylogenetic least squares models for shape and other high-dimensional multivariate data. Evolution. 68. DOI:10.1111/evo.12463.

Examples

```
### Example of D-PGLS for high-dimensional data
data(plethspecies)
Y.gpa<-gpagen(plethspecies$land)    #GPA-alignment

procD.pgls(two.d.array(Y.gpa$coords)~Y.gpa$Csize,plethspecies$phy,iter=49)
```

---

ratland *Landmark data from dataset rat*

Description

Landmark data from dataset rat

Author(s)

Dean Adams

References

Bookstein, F. L. 1991. Morphometric tools for landmark data: Geometry and Biology. Cambridge Univ. Press, New York.

`read.morphologika` *Read landmark data from Morphologika file*

Description

Read Morphologika file (*.txt) to obtain landmark coordinates and specimen information

Usage

```
read.morphologika(file)
```

Arguments

`file`	A Morphologika *.txt file containing two- or three-dimensional landmark data.

Details

This function reads a *.txt file in the Morphologika format containing two- or three-dimensional landmark coordinates. Morphologika files are text files in one of the standard formats for geometric morphometrics (see O'Higgins and Jones 1998,2006).

If the headers "[labels]" and "[labelvalues]" are present, then a data matrix containing all individual specimen information is returned. If the header "[wireframe]" is present, then a matrix of the landmark addresses for the wireframe is returned (see `plotRefToTarget` option 'links')

Value

Function returns a (p x k x n) array of the coordinate data. If other optional headers are present in the file (e.g. "[labels]" or "[wireframe]") function returns a list containing the "coords" array, and data matrix of "labels" and or "wireframe".

Author(s)

Emma Sherratt & Erik Otarola-Castillo

References

O'Higgins P and Jones N (1998) Facial growth in Cercocebus torquatus: An application of three dimensional geometric morphometric techniques to the study of morphological variation. Journal of Anatomy. 193: 251-272

O'Higgins P and Jones N (2006) Tools for statistical shape analysis. Hull York Medical School.

read.ply *Read mesh data (vertices and faces) from ply files*

Description

A function to read ply files, which can be used for digitizing landmark coordinates or for shape warps.

Usage

```
read.ply(file, ShowSpecimen = TRUE)
```

Arguments

file	An ASCII ply file
ShowSpecimen	logical A logical value indicating whether or not the ply file should be displayed

Details

Function reads three-dimensional surface data in the form of a single ply file (Polygon File Format; ASCII format only, from 3D scanners such as NextEngine and David scanners). Vertices of the surface may then be used to digitize three-dimensional points, and semilandmarks on curves and surfaces. The surface may also be used as a mesh for visualizing 3D deformations (warpRefMesh). The function opens the ply file and plots the mesh, with faces rendered if file contains faces, and colored if the file contains vertex color.

Value

Function returns the following components:

mesh3d	list of class mesh3d- see rgl for details

Author(s)

Dean Adams & Emma Sherratt

Examples

```
# If the file has no mesh color, or color is undesirable, user can assign this as follows:
# Using the example scallop PLY
data(scallopPLY)
myply <- scallopPLY$ply
myply$material <- "gray" # using color word
myply$material <- "#FCE6C9" # using RGB code
```

`readland.nts` *Read landmark data matrix from nts file*

Description

Read *.nts file to obtain landmark coordinates for a set of specimens

Usage

```
readland.nts(file)
```

Arguments

`file`	A *.nts file containing two- or three-dimensional landmark data

Details

Function reads a *.nts file containing a matrix of two- or three-dimensional landmark coordinates. NTS files are text files in one of the standard formats for geometric morphometrics (see Rohlf 2012). The parameter line contains 5 or 6 elements, and must begin with a "1" to designate a rectangular matrix. The second and third values designate how many specimens (n) and how many total variables (p x k) are in the data matrix. The fourth value is a "0" if the data matrix is complete and a "1" if there are missing values. If missing values are present, the '1' is followed by the arbitrary numeric code used to represent missing values (e.g., -999). These values will be replaced with "NA" in the output array. Subsequent analyses requires a full complement of data, see `estimate.missing`. The final value of the parameter line denotes the dimensionality of the landmarks (2,3) and begins with "DIM=". If specimen and variable labels are included, these are designated placing an "L" immediately following the specimen or variable values in the parameter file. The labels then precede the data matrix.

Missing data may also be represented by designating them using 'NA'. In this case, the standard NTSYS header is used with no numeric designation for missing data (i.e. the fourth value is '0'). The positions of missing landmarks may then be estimated using estimate.missing.

Function is for *.nts file containing landmark coordinates for multiple specimens. Note that *.dta files in the nts format written by Landmark Editor http://graphics.idav.ucdavis.edu/research/projects/EvoMorph, and *.nts files written by Stratovan Checkpoint http://www.stratovan.com/ have incorrect header notation; every header is 1 n p-x-k 1 9999 Dim=3, rather than 1 n p-x-k 0 Dim=3, which denotes that missing data is in the file even when it is not.

Value

Function returns a (p x k x n) array, where p is the number of landmark points, k is the number of landmark dimensions (2 or 3), and n is the number of specimens. The third dimension of this array contains names for each specimen, which are obtained from the image names in the *.nts file.

Author(s)

Dean Adams & Emma Sherratt

References

Rohlf, F. J. 2012 NTSYSpc: Numerical taxonomy and multivariate analysis system. Version 2.2. Exeter Software, New York.

`readland.tps` *Read landmark data from tps file*

Description

Read *.tps file to obtain landmark coordinates

Usage

```
readland.tps(file, specID = c("None", "ID", "imageID"), warnmsg = T)
```

Arguments

`file`	A *.tps file containing two- or three-dimensional landmark data
`specID`	a character specifying whether to extract the specimen ID names from the ID or IMAGE lines (default is "None").
`warnmsg`	A logical value stating whether warnings should be printed

Details

This function reads a *.tps file containing two- or three-dimensional landmark coordinates. Tps files are text files in one of the standard formats for geometric morphometrics (see Rohlf 2010). Two-dimensional landmarks coordinates are designated by the identifier "LM=", while three-dimensional data are designated by "LM3=". Landmark coordinates are multiplied by their scale factor if this is provided for all specimens. If one or more specimens are missing the scale factor, landmarks are treated in their original units.

Missing data may be present in the file. In this case, they must be designated by 'NA'. The positions of missing landmarks may then be estimated using estimate.missing.

The user may specify whether specimen names are to be extracted from the 'ID=' field or 'IMAGE=' field and included in the resulting 3D array. e.g., for 'ID=' use (file, specID = "ID") and for 'IMAGE=' use (file, specID = "imageID"). The default is specID="None".

NOTE: At present, all other information that can be contained in tps files (curves, comments, variables, radii, etc.) is ignored.

Value

Function returns a (p x k x n) array, where p is the number of landmark points, k is the number of landmark dimensions (2 or 3), and n is the number of specimens. The third dimension of this array contains names for each specimen, which are obtained from the image names in the *.tps file.

Author(s)

Dean Adams & Emma Sherratt

References

Rohlf, F. J. 2010. tpsRelw: Relative warps analysis. Version 1.49. Department of Ecology and Evolution, State University of New York at Stony Brook, Stony Brook, NY.

`readmulti.nts` *Read landmark data from multiple nts files*

Description

Read a list of names for several *.nts files to obtain landmark coordinates for a set of specimens

Usage

```
readmulti.nts(filelist)
```

Arguments

`filelist`	A list of names for the *.nts files to be read by the function. The names in the list require quotes (") and .nts/.NTS suffix.

Details

This function reads a list containing the names of multiple *.nts files, where each contains the landmark coordinates for a single specimen. For these files, the number of variables (columns) of the data matrix will equal the number of dimensions of the landmark data (k=2 or 3). When the function is called a dialog box is opened, from which the user may select multiple *.nts files. These are then read and concatenated into a single matrix for all specimens. The parameter line contains 5 or 6 elements, and must begin with a "1" to designate a rectangular matrix. The second and third values designate how many specimens (n) and how many total variables (p x k) are in the data matrix. The fourth value is a "0" if the data matrix is complete and a "1" if there are missing values. If missing values are present, the '1' is followed by the arbitrary numeric code used to represent missing values (e.g., -999). These values will be replaced with "NA" in the output array. Subsequent analyses requires a full complement of data, see `estimate.missing`.

Missing data may be present in the file by designating them using 'NA'. In this case, the standard NTSYS header is used with no numeric designation for missing data (i.e. the fourth value is '0'). The positions of missing landmarks may then be estimated using estimate.missing.

Value

Function returns a (p x k x n) array, where p is the number of landmark points, k is the number of landmark dimensions (2 or 3), and n is the number of specimens. The third dimension of this array contains names for each specimen, which are obtained from the original file names.

Author(s)

Dean Adams & Emma Sherratt

`scallopPLY` *3D scan of a scallop shell from a .ply file in mesh3d format*

Description

3D scan of a scallop shell from a .ply file in mesh3d format

Author(s)

Emma Sherratt

References

Serb et al. (2011). "Morphological convergence of shell shape in distantly related scallop species (Mollusca: Pectinidae)." Zoological Journal of the Linnean Society 163: 571-584.

`scallops` *Landmark data from scallop shells*

Description

Landmark data from scallop shells

Author(s)

Dean Adams and Erik Otarola-Castillo

References

Serb et al. (2011). "Morphological convergence of shell shape in distantly related scallop species (Mollusca: Pectinidae)." Zoological Journal of the Linnean Society 163: 571-584.

`trajectory.analysis` *Quantify and compare shape change trajectories*

Description

Function estimates attributes of shape change trajectories or motion trajectories for a set of Procrustes-aligned specimens and compares them statistically

Usage

```
trajectory.analysis(f1, data = NULL, estimate.traj = TRUE,
  traj.pts = NULL, iter = 99)
```

Arguments

`f1`	A formula for the linear model (e.g., y~x1+x2)
`data`	An optional value specifying a data frame containing all data (not required)
`estimate.traj`	A logical value indicating whether trajectories are estimated from original data; described below
`iter`	Number of iterations for significance testing
`traj.pts`	An optional value specifying the number of points in each trajectory (if estimate.traj=FALSE)

Details

The function quantifies phenotypic shape change trajectories from a set of specimens, and assesses variation in these parameters via permutation. A shape change trajectory is defined by a sequence of shapes in tangent space. These trajectories can be quantified various attributes (their size, orientation, and shape), and comparisons of these attribute enables the statistical comparison of shape change trajectories (see Collyer and Adams 2013; Collyer and Adams 2007; Adams and Collyer 2007; Adams and Collyer 2009).

Data input is specified by a formula (e.g., Y~X), where 'Y' specifies the response variables (trajectory data), and 'X' contains one or more independent variables (discrete or continuous). The response matrix 'Y' must be in the form of a two-dimensional data matrix of dimension (n x [p x k]), rather than a 3D array. The function `two.d.array` can be used to obtain a two-dimensional data matrix from a 3D array of landmark coordinates. It is assumed that the order of the specimens 'Y' matches the order of specimens in 'X'.

There are two primary modes of analysis through this function. If "estimate.traj=TRUE" the function estimates shape trajectories using the least-squares means for groups, based on a two-factor model (e.g., Y~A+B+A:B). Under this implementation, the last factor in 'X' must be the interaction term, and the preceding two factors must be the effects of interest. Covariates may be included in 'X', and must precede the factors of interest (e.g., Y~cov+A*B). In this implementation, 'Y' contains a matrix of landmark coordinates. It is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with `gpagen`].

If "estimate.traj=FALSE" the trajectories are assembled directly from the set of shapes provided in 'Y'. With this implementation, the user must specify the number of shapes that comprise each trajectory. This approach is useful when the set of shapes forming each trajectory have been quantified directly (e.g., when motion paths are compared: see Adams and Cerney 2007). With this implementation, variation in trajectory size, shape, and orientation are evaluated for each term in 'X'.(see Adams and Cerney 2007).

Once the function has performed the analysis, it generates a plot of the trajectories as visualized in the space of principal components (PC1 vs. PC2). The first point in each trajectory is displayed as white, the last point is black, and any middle points on the trajectories are in gray. The colors of trajectories follow the order in which they are found in the dataset, using R's standard color palette: black, red, green3, blue, cyan, magenta, yellow, and gray.

Value

If "estimate.traj=TRUE", the function returns a list with the following components:

`procDist.lm`	Procrustes ANOVA table
`traj.size`	A matrix of pairwise differences in trajectory size
`p.size`	A matrix of pairwise significance levels for trajectory size
`traj.orient`	A matrix of pairwise differences in trajectory orientation
`p.orient`	A matrix of pairwise significance levels for trajectory orientation
`traj.shape`	A matrix of pairwise differences in trajectory shape (if applicable)
`p.shape`	A matrix of pairwise significance levels for trajectory shape

If "estimate.traj=FALSE", the function returns a list with the following components:

`MANOVA.location.covariation`	Procrustes ANOVA table
`ANOVA.Size`	Results of permutational-ANOVA assessing variation in trajectory size
`ANOVA.Dir`	Results of permutational-ANOVA assessing variation in trajectory orientation
`ANOVA.Shape`	Results of permutational-ANOVA assessing variation in trajectory shape (if applicable)

Author(s)

Dean Adams

References

Collyer, M.L., and D.C. Adams. 2013. Phenotypic trajectory analysis: Comparison of shape change patterns in evolution and ecology. Hystrix. 24:75-83.

Adams, D. C. 2010. Parallel evolution of character displacement driven by competitive selection in terrestrial salamanders. BMC Evol. Biol. 10:1-10.

Adams, D. C., and M. M. Cerney. 2007. Quantifying biomechanical motion using Procrustes motion analysis. J. Biomech. 40:437-444.

Adams, D. C., and M. L. Collyer. 2007. The analysis of character divergence along environmental gradients and other covariates. Evolution 61:510-515.

Adams, D. C., and M. L. Collyer. 2009. A general framework for the analysis of phenotypic trajectories in evolutionary studies. Evolution 63:1143-1154.

Collyer, M. L., and D. C. Adams. 2007. Analysis of two-state multivariate phenotypic change in ecological studies. Ecology 88:683-692.

Examples

```
#1: Estimate trajectories from LS means in 2-factor model
data(plethodon)
Y.gpa<-two.d.array(gpagen(plethodon$land)$coords)

trajectory.analysis(Y.gpa~plethodon$species*plethodon$site,iter=15)

#2: Compare motion trajectories
data(motionpaths)

#Motion paths represented by 5 time points per motion

trajectory.analysis(motionpaths$trajectories~motionpaths$groups,
estimate.traj=FALSE, traj.pts=5,iter=15)
```

---

two.b.pls *Two-block partial least squares analysis for shape data*

Description

Function performs two-block partial least squares analysis to assess the degree of association between to blocks of Procrustes-aligned coordinates (or other variables)

Usage

```
two.b.pls(A1, A2, warpgrids = TRUE, iter = 999, verbose = FALSE,
  label = NULL)
```

Arguments

A1	A matrix (n x [p x k]) or 3D array (p x k x n) containing GPA-aligned coordinates for the first block
A2	A matrix (n x [p x k]) or 3D array (p x k x n) containing GPA-aligned coordinates for the second block
iter	Number of iterations for significance testing
label	An optional vector indicating labels for each specimen that are to be displayed
warpgrids	A logical value indicating whether deformation grids for shapes along PC1 should be displayed (only relevant if data for A1 or A2 [or both] were input as 3D array)
verbose	A logical value indicating whether the output is basic or verbose (see Value below)

Details

The function quantifies the degree of association between two blocks of shape data as defined by landmark coordinates using partial least squares (see Rohlf and Corti 2000). If geometric morphometric data are used, it is assumed that the landmarks have previously been aligned using Generalized Procrustes Analysis (GPA) [e.g., with gpagen]. If other variables are used, they must be input as a 2-Dimensional matrix (rows = specimens, columns = variables).

A plot of PLS scores from Block1 versus Block2 is provided for the first set of PLS axes. Thin-plate spline deformation grids along these axes are also shown (if data were input as a 3D array).

Value

Function returns a list with the following components:

value	The estimate of association between block
pvalue	The significance level of the observed association
Xscores	PLS scores for the first block of landmarks (when verbose=TRUE)
Yscores	PLS scores for the second block of landmarks (when verbose=TRUE)

Author(s)

Dean Adams

References

Rohlf, F.J., and M. Corti. 2000. The use of partial least-squares to study covariation in shape. Systematic Biology 49: 740-753.

Examples

```
data(plethShapeFood)
Y.gpa<-gpagen(plethShapeFood$land)    #GPA-alignment

#2B-PLS between head shape and food use data
two.b.pls(Y.gpa$coords,plethShapeFood$food,iter=99)
```

---

two.d.array *Convert (p x k x n) data array into 2D data matrix*

Description

Convert a three-dimensional array of landmark coordinates into a 2-dimensional matrix

Usage

```
two.d.array(A)
```

Arguments

A	An array (p x k x n) containing landmark coordinates for a set of specimens

Details

This function converts a (p x k x n) array of landmark coordinates into a 2-dimensional matrix. The latter format of the shape data is useful for performing subsequent statistical analyses in R (e.g., PCA, MANOVA, PLS, etc.). Row labels are preserved if included in the original array.

Value

Function returns a two-dimensional matrix of dimension (n x [p x k]), where rows represent specimens and columns represent variables.

Author(s)

Dean Adams and Emma Sherratt

Examples

```
data(plethodon)
plethodon$land    #original data in the form of 3D array

two.d.array(plethodon$land)   # Convert to a 2D data matrix
```

---

warpRefMesh *Creates a mesh3d object warped to the mean shape.*

Description

A function to take a ply file and use thin-plate spline method to warp the file into the estimated mean shape for a set of aligned specimens.

Usage

```
warpRefMesh(file, mesh.coord, ref, color = NULL, centered = FALSE)
```

Arguments

file	An ASCII ply file
mesh.coord	A p x k matrix of 3D coordinates digitized on the ply file.
ref	A p x k matrix of 3D coordinates made by mshape
color	Color to set the ply file $material. If the ply already has color, use NULL. For ply files without color, color=NULL will be plotted as grey.
centered	Logical If the data in mesh.coords were collected from a centered mesh (see details).

Details

Function takes a ply file and its landmark coordinates uses the thin-plate spline method (Bookstein 1989) to warp the mesh into the shape defined by a second set of landmark coordinates, usually those of the mean shape for a set of aligned specimens. It is highly recommended that the mean shape is used as the reference for warping (see Rohlf 1998). The workflow is as follows:

1. Calculate the mean shape using `mshape`
2. Choose an actual specimen to use for the warping. The specimen used as the template for this warping is recommended as one most similar in shape to the average of the sample, but can be any reasonable specimen - do this by eye, or use `findMeanSpec`
3. Warp this specimen into the mean shape using `warpRefMesh`
4. Use this average mesh where it asks for a mesh= in the analysis functions and visualization functions

For landmark coordinates digitized with geomorph digitizing functions, centered = TRUE. This refers to the specimen being centered prior to landmark acquisition in the RGL window. For landmark data collected outside of geomorph, centered=FALSE will usually be the case. The returned mesh3d object is for use in geomorph functions where shape deformations are plotted (`plotTangentSpace`, `plotAllometry`, `bilat.symmetry`, and `plotRefToTarget`).

Value

Function returns a mesh3d object, which is a list of class mesh3d (see rgl for details)

Author(s)

Emma Sherratt

References

Bookstein, F. L. 1989 Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(6):567-585.

Rohlf, F. J. 1998. On Applications of Geometric Morphometrics to Studies of Ontogeny and Phylogeny. Systematic Biology. 47:147-158.

See Also

`findMeanSpec`

`writeland.tps` *Write landmark data to tps file*

Description

Write *.tps file from obtain landmark coordinates in a 3-dimensional array

Usage

```
writeland.tps(A, file)
```

Arguments

`A`	An array (p x k x n) containing landmark coordinates for a set of specimens
`file`	Name of the *.tps file to be created

Details

This function writes a *.tps file from a 3-dimensional array (p x k x n) of landmark coordinates.

Author(s)

Dean Adams

Index