Package ‘moRphomenses’

May 9, 2026

Title Geometric Morphometric Tools to Align, Scale, and Compare ``Shape'' of Menstrual Cycle Hormones

Version 1.0.3

Description Mitteroecker & Gunz (2009) <doi:10.1007/s11692-009-9055-x> describe how geometric morphometric methods allow researchers to quantify the size and shape of physical biological structures. We provide tools to extend geometric morphometric principles to the study of non-physical structures, hormone profiles, as outlined in Ehrlich et al (2021) <doi:10.1002/ajpa.24514>. Easily transform daily measures into multivariate landmark-based data. Includes custom functions to apply multivariate methods for data exploration as well as hypothesis testing. Also includes 'shiny' web app to streamline data exploration. Developed to study menstrual cycle hormones but functions have been generalized and should be applicable to any biomarker over any time period.

License GPL (>= 3.0)

URL <`https://github.com/ClancyLabUIUC/moRphomenses`>

Depends R (>= 2.10)

Imports stats, graphics, grDevices, utils

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests dendextend, geomorph, cluster, shiny

NeedsCompilation no

Author Daniel Ehrlich [aut, cre]

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

Repository CRAN

Date/Publication 2025-01-08 18:30:02 UTC

Contents

`mm_ArrayData` *Array Data*

Description

Construct a ragged array (containing missing data) of a specified length (up/down sampling individuals to fit).

Usage

```
mm_ArrayData(
  IDs,
  DAYS,
  VALUE,
  MID = NULL,
```

```
  targetLENGTH,
  targetMID = NULL,
  transformation = c("minmax", "geom", "zscore", "log", "log10"),
  impute_missing = 3
)
```

Arguments

`IDs`	A vector that contains individual IDs repeated for multiple days of collection.
`DAYS`	A vector that contains information on time, IE Day 1, Day 2, Day 3. Note: this vector should include integers, continuous data might produce unintended results.
`VALUE`	A vector containing the variable sampled.
`MID`	Am optional vector of midpoints to center each individuals profile. These should be unique to each individual and repeated for each observation of DAYS, VALUE, and IDs. If NULL (defualt), data will not be centered on any day.
`targetLENGTH`	Integer. Number of days to up/down sample observations to using `mm_get_interval`.
`targetMID`	If NULL (default) data will not be centered and will range from 0 to 1. If specified, data will be centered on 0 ranging from -1 to 1.
`transformation`	Which (if any) data transformation to apply. Our reccomendation is minmax, but Geometric mean, Zscore, natural log and log10 transformations are available, if desired.
`impute_missing`	Integer. If not null, number of nearest-neighbors to use to impute missing data (Default = 3).

Value

Returns a 3D array of data to be analyzed with individuals in the 3rd dimension.

`mm_BuildModel` *Build, implement, visualize multivariate linear model.*

Description

Easily evaluate simple model sets (one covariate with up to 2 additional classifiers/covariates). Helpful for exploratory analysis. For detailed models or specific combinations of variables, see geomorph::procD.lm for full use of this function.

Usage

```
mm_BuildModel(shape_data, ..., subgrps = NULL, ff1 = NULL, univ_series = FALSE)
```

Arguments

`shape_data`	This will be the (multivariate) response variable
`...`	Covariate(s)/classifier(s) to build a model set. Individual models are run with interaction effects.
`subgrps`	Optional. Vector of group membership. Model sets will be run across the whole sample and subgroups. If k is specified, only the full model will be run.
`ff1`	An explicit model to test in the format: " coords ~ ...". Names must match those specifed in `...`. Standard lm notation applies.
`univ_series`	Default (FALSE) will evaluate multiple covariates and their interaction in a single model. However, it can be helpful to understand the univariate effects in isolation of interaction/confounding factors. Set `univ_series=TRUE` to produce a series of model sets, one for each covariate specified (NOTE: ff1 must also be NULL for this to work).

Value

A list containing output of one or more multivariate linear models that can be inspected on their own or interacted with using mm_VizModel or mm_CompModel.

`mm_CalcShapespace` *Define Shapespace of aligned dataset.*

Description

Conduct PCA of shape data and visualize major shape trends.

Usage

```
mm_CalcShapespace(dat, max_Shapes = 10)
```

Arguments

`dat`	A 3D array of shape data to be analyzed.
`max_Shapes`	The maximum amount of PCs to visualize. Default 10.

Value

A list containing the results of shape-pca, including vizualizations of shape extrema for each Principal Component.

`mm_CheckImputation` *Check Imputation*

Description

Plot Raw (aligned) data along side by side with imputed data.

Usage

```
mm_CheckImputation(A1, A2, ObO = interactive())
```

Arguments

`A1`	An aligned array, containing missing data (presumably made with `mm_ArrayData$shape_data_wNA`).
`A2`	An aligned and imputed array (presumably made with `mm_ArrayData$Shape_data`).
`ObO`	One-by-One. If TRUE (default, in interactive sessions), individuals will be plotted one at a time, requiring the user to advance/exit the operation. If FALSE, all plots #' will be generated at once to be browsed or exported from the `Plots` panel.

Value

A series of plots for each individual in the array. If `ObO=TRUE` user input is required to advance or exit the plotting.

`mm_ColorLeaves` *Color leves of a dendrogram*

Description

Specify color order approriately for a dendrogram

Usage

```
mm_ColorLeaves(dendro, cols)
```

Arguments

`dendro`	A dendrogram or hclust class object
`cols`	a vector of colors

Details

Leaves of a dendrogram will be re-ordered compared to most input classifiers. This function takes the study-ordered colors and correctly applies them to the dendrogram using `dendextend`

Value

A dendrogram class object with leaves colored as specified.

`mm_CompModel` *Compare Model Metrics*

Description

Compare key figs (Rsq, p-value, etc) across multiple models.

Usage

```
mm_CompModel(mv_results, row_labels = NULL, digits = 4)
```

Arguments

`mv_results`	Input mvlm, created by mm_BuildModel (or by using geomorph::procD.lm)
`row_labels`	A character vector to use in output. If NULL (default) labels from the input data will be used.
`digits`	Number of decimal places to round to. Default includes 4 decimal places.

Value

A list containing the results of the mvlm, visualizations of shape trends along the regression line, and the model itself.

`mm_CompModel_Full` *Compare Complex Model Metrics*

Description

Compare key figs (Rsq, p-value, etc) across multiple complex models.

Usage

```
mm_CompModel_Full(mv_results, row_labels = NULL, var_labels = NULL, digits = 4)
```

Arguments

`mv_results`	Input mvlm, created by mm_BuildModel (or by using geomorph::procD.lm)
`row_labels`	A character vector to use in output. If NULL (default) labels from the input data will be used.
`var_labels`	A character vector to use in output. If NULL (default) labels from the input data will be used.
`digits`	Number of decimal places to round to. Default includes 4 decimal places.

Value

description

`mm_coords_to_shape` *Visualize shape of target coordinates*

Description

Visualize shape of target coordinates

Usage

```
mm_coords_to_shape(A, PCA, target_coords, target_PCs = c(1, 2))
```

Arguments

`A`	A landmark array used for the pca
`PCA`	output of prcomp. Should contain $transormation
`target_coords`	A single set of X,Y coordinates.
`target_PCs`	Integer identifying which pc to use on the X and Y axis. Default is c(1,2) for PC1 on x and PC2 on y

Value

A landmark array representing the hypothetical shape of a given set of coordinates.

`mm_data` *Sample hormone dataset*

Description

Sample dataset classifiers to be paired with sample array. This table contains 60 rows to match the 60 individuals across the third dimension of the array

Usage

```
mm_data
```

Format

A matrix with 2015 obs (rows) and 4 variables (columns).

ID Individual id, each integer represents a different individual.

CYCLEDAY Integer day of cycle. Generally runs from 1 ... (28 on average).

MIDPOINT Single value for each individual, repeated along each CYCLEDAY. In this sample, day of ovulation.

E1G Daily measure of hormone, in nanograms per mililiter

`mm_Diagnostics` *Run a suite of diagnostic analyses.*

Description

Conduct a set of analyses to make shape-PCA results easier to interpret. Specifically, this will provide a table of eigen values (optional barplot), provide 5-number summary across each PC, conduct a naive Ward's clustering of PC scores (optional dendrogram, along with silhouette plot and scree plot of individual distance to the sample mean

Usage

```
mm_Diagnostics(dat, max_PC_viz = 10, max_PC_calc = NULL, hide_plots = FALSE)
```

Arguments

`dat`	A 3D array or a mmPCA object (output of mm_CalcShapespace).
`max_PC_viz`	Maximum number of PCs to include in visualizations (EG Eigenplots, or shape trends.
`max_PC_calc`	By default (NULL), all PCs will be included in calculations. However, if fewer PCs are required users may specify an integer, n, to get the first n PCS.
`hide_plots`	By default (FALSE), helpful visuals are plotted.

Value

Returns a list containing the results of:

- eigs - A table containing individual and cumulutive loadings for each PC
- PC_5_num - A data.frame containing the fivenum summary for each PC
- TREE - A dendrogram representing the results of a naive-Ward's clustering

---

`mm_ellipse` *Add confidence ellipses to an active scatterplot.*

Description

Add confidence ellipses to an active scatterplot.

Usage

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

Arguments

`dat`	A matrix of data to draw an ellipses around.
`ci`	Percentage of data to capture. Must be one of c(67.5, 90, 95, 99).
`linesCol`	Border color of the shape.
`fillCol`	Fill color of the shape.
`smoothness`	Lower values will look jagged, higher value will make smoother lines, but may take a long time to plot. Default value is 20.

Value

No value. Will add an ellipses of a given size to the current plot.

`mm_Explorer` *Launch mm_Explorer*

Description

Launch mm_Explorer

Usage

```
mm_Explorer()
```

Value

No value. Will launch `shiny` app in default web browser.

`mm_FillMissing` *Impute Missing Data*

Description

Fill in a ragged away by nearest neighbor imputation

Usage

```
mm_FillMissing(A, knn = 3)
```

Arguments

`A`	A ragged array (IE, contains missing cells), presumably constructed with `mm_ArrayData`.
`knn`	Number of nearest neighbors to draw on for imputation (default = 3).

Value

Returns an array of the same dimensions with all missing data filled.

`mm_FlattenArray` *Flatten Array*

Description

Convert a 3D array to 2D matrix suitable for PCA, etc. Note, this function is identical to geomorph::two.d.array, reproduced here for convenience.

Usage

```
mm_FlattenArray(A, sep = ".")
```

Arguments

`A`	an array to be flattened
`sep`	Separator to be used for column names

Value

Returns a flattened array

`mm_get_interval` *Create equallly spaced intervals.*

Description

Create a sequence from -1:1 of specified length. MIDpoint (day0) can be

Usage

```
mm_get_interval(days, day0 = NULL)
```

Arguments

`days`	The length of the sequence to return, inclusive of the endpoints (-1,1)
`day0`	If NULL (default), the median integer will be calculated, centering the range on 0. Specifying a value will set 0 to that value, creating asymmetric ranges.

Value

Returns a numeric vector of specified length, ranging from -1 to 1

Examples

```
mm_get_interval(15) ## Symmetrical sequence from -1 to 1 with 0 in the middle.
mm_get_interval(15, day0 = 8) ## The same sequence, explicitly specifying the midpoint
mm_get_interval(15, day0 = 3) ## 15 divisions with an asymmetric distribution.
```

`mm_grps_PlotArray` *Plot Arrays of groups*

Description

Attempts to optimally format a grid of arrays by group

Usage

```
mm_grps_PlotArray(A, grps, reset_par = TRUE)
```

Arguments

`A`	an array to be plotted
`grps`	a vector defining group IDs to subset along the 3rd dimension of the array
`reset_par`	Optional, default = TRUE. If false, do not reset graphic parameters in order to create complex plots.

Details

4 Groups will plot as a 2x2 grid, while 9 groups plot in a 3x3. Function is experimental

Value

Returns no values, produces a series of plots.

`mm_grp_dists` *Distance from Centroid*

Description

Calculate and plot group distance from centroid (grand mean)

Usage

```
mm_grp_dists(dat, grps, plots = TRUE)
```

Arguments

`dat`	a 2d matrix of data. Presumably PC scores
`grps`	a vector defining group IDs
`plots`	Logical. Should distances be plotted as boxplots? If FALSE, distance calculations are still performed

Value

A list containing individual distances from the sample mean shape. If `plots=TRUE`, will also visualize results

---

`mm_mute_cols` *Take a color and modify it*

Description

Modify color/transparency using hsv syntax

Usage

```
mm_mute_cols(cols, s = NULL, v = NULL, alpha = 0.4)
```

Arguments

`cols`	a vector of colors, eg: "#0066FF"
`s`	Either a single value or a vector of same length as cols specifying a new saturation (range 0-1). colors darken to black (0).
`v`	Either a single value or a vector of same length as cols specifying a new value (range 0-1). colors lighten to white (0)
`alpha`	Either a single value or a vector of same length as cols specifying a transparency value (range 0-1). colors translucent at 0.

Value

A vector of colors that have been modified in saturation, value, or alpha

---

`mm_Phenotype` *Generate Phenotypes*

Description

Partition sample into clusters, based on information from

Usage

```
mm_Phenotype(dat, kgrps, cuttree_h = NULL, cuttree_k = NULL, plot_figs = TRUE)
```

Arguments

`dat`	Either an Array of shape data, an mmPCA object, or an mmDiag object.
`kgrps`	A non-negative integer of sub-groups to draw. kgrps=1 will provide results for the whole input dat.
`cuttree_h`	Optional. Draw clusters by splitting the tree at a given height, h.
`cuttree_k`	Optional. Draw clsuters by splitting the tree into number of branches, k
`plot_figs`	Optional. Default = TRUE, plot phenotypes for each set(s) of subgroups.

Value

If plot_figs=TRUE (Default), plot associated graphs and return a list containing:

- ALN - an array containing aligned and scaled landmark data, the output of mm_ArrayData
- PCA - PC scores, eigenvalues, and shape visualizations, the output of mm_CalcShapespace
- TREE - Dendrogram of PC scores, the output of mm_Diagnostics
- k_grps - If `kgrps` is specified, a vector defining group membership (as integer); the results of k-means clustering based on PC scores.
- cth_grps - If `cth_grps` is specified, a vector defining group membership (as integer); the results of clustering using stats::cutree for a given height.
- ctk_grps - If `ctk_grps` is specified, a vector defining group membership (as integer); the results of clustering using stats::cutree for a given number of clusters.

`mm_PlotArray` *Plot Array Plot individuals and optionally mean form*

Description

Plot Array Plot individuals and optionally mean form

Usage

```
mm_PlotArray(
  A,
  MeanShape = TRUE,
  AllCols = NULL,
  MeanCol = NULL,
  plot_type = c("lines", "points"),
  lbl = NULL,
  yr = NULL,
  axis_labels = FALSE,
  reset_par = TRUE
)
```

Arguments

`A`	An array to be plotted
`MeanShape`	Logical. Should the Mean Shape be calculated and plotted
`AllCols`	Either a single color for all individuals, or a vector specifying colors for each individual. If NULL (default) individuals will be plotted in grey
`MeanCol`	A single color for the mean shape. If Null (default) mean shape will be plotted in black
`plot_type`	Should the data be plotted as points or lines.
`lbl`	A title (main =) for the plot. If NULL (default) the name of the array will be used.
`yr`	Y-range, in the form c(0,100)
`axis_labels`	Should units be printed along the axis. Defaults to FALSE to maximize the profile shape.
`reset_par`	Optional, default = TRUE. If false, do not reset graphic parameters in order to create complex plots.

Value

Plot individual(s) profile(s) in the default graphics device.

`mm_pretty_pca` *Plot Calendar Days*

Description

Pretty PCA

Usage

```
mm_pretty_pca(PCA, xPC = 1, yPC = 2, clas_col = NULL, legend_cex = 0.8)
```

Arguments

`PCA`	Input data either prcomp or mmPCA.
`xPC`	The PC to plot on the x axis
`yPC`	The PC to plot on the y axis
`clas_col`	A character vector of groupings. Each level will be plotted as a different color.
`legend_cex`	A scaling factor to be applied specifically to the legend. Set to NULL for scatterplot only.

Details

A better PCA plot

Value

Returns no object, plots results of PCA

`mm_ScreePlot` *Scree Plot*

Description

Plot total within group sum of squares to evalaute clusters

Usage

```
mm_ScreePlot(x, maxC = 15, ...)
```

Arguments

`x`	Input data for cluster analysis (IE, PCA)
`maxC`	Maximum clusters to evaluate
`...`	Additional arguments to be passed to plot

Value

No value, produces diagnostic plot.

`mm_SilPlot` *Silhouette Width Plot*

Description

Plot average silhouete widths to evaluate clusters

Usage

```
mm_SilPlot(x, maxC = 15, ...)
```

Arguments

`x`	Input data for cluster analysis (IE PCA)
`maxC`	Maximum clusters to evaluate
`...`	additional arguments passed to plot

Value

No value, produces diagnostic plot.

`mm_transf_geom` *Geometric Scaling*

Description

Calculate the geometric mean of a vector and scale all values by it.

Usage

```
mm_transf_geom(x)
```

Arguments

`x`	A numeric vector to be scaled. Missing values will produce NA, conduct knn imputation using mm_FillMissing first.

Value

Returns a scaled vector

Examples

```
mm_transf_geom(1:10)
```

`mm_transf_log` *natural log transform*

Description

Transform a vector by the natural log.

Usage

```
mm_transf_log(x)
```

Arguments

`x`	A numeric vector to be scaled. Missing values will produce NA, conduct knn imputation using mm_FillMissing first.

Value

Returns a scaled vector

Examples

```
mm_transf_log(1:10)
```

`mm_transf_log10` *Common log transform*

Description

Transform a vector by the common log (base 10).

Usage

```
mm_transf_log10(x)
```

Arguments

`x`	A numeric vector to be scaled. Missing values will produce NA, conduct knn imputation using mm_FillMissing first.

Value

Returns a scaled vector

Examples

```
mm_transf_log10(1:10)
```

`mm_transf_minmax` *Min-Max Scaling*

Description

Scale a vector from 0,1 based on its minimum and maximum values.

Usage

```
mm_transf_minmax(x)
```

Arguments

`x`	A Numeric vector to be scaled. Missing values are allowed and ignored.

Value

Returns a scaled vector

Examples

```
mm_transf_minmax(1:10)
```

`mm_transf_zscore` *Z scores*

Description

Calculate and return z-scores given a numeric vector.

Usage

```
mm_transf_zscore(x)
```

Arguments

`x`	A numeric vector to be scaled. Missing values will produce NA, conduct knn imputation using mm_FillMissing first.

Value

Returns a scaled vector

Examples

```
mm_transf_zscore(1:10)
```

`mm_VizModel` *Visualize Multivariate LM*

Description

Visualize 2D scatterplot of mvlm including predicted shapes.

Usage

```
mm_VizModel(dat, clas_col = NULL)
```

Arguments

`dat`	Input mvlm, created by mm_BuildModel (or by using geomorph::procD.lm)
`clas_col`	A classifier to color the data by. If null (default) all points will be grey. Otherwise, data will be plotted as rainbow(n) colors.

Value

A list containing the results of the mvlm, visualizations of shape trends along the regression line, and the model itself.

`mm_VizShapespace` *Visualize PC axes*

Description

Plot a scatterplot and vizualize shape change across the X axis.

Usage

```
mm_VizShapespace(
  mmPCA,
  xPC = 1,
  yPC = 2,
  yr = c(0, 1.1),
  cols = NULL,
  title = "",
  png_dir = NULL
)
```

Arguments

`mmPCA`	Output of `mm_CalcShapespace`, containing a PCA object with PC shapes
`xPC`	The PC to be plotted on the x axis. If yPC is left null, a univariate density distribution will be plotted with min/max shapes.
`yPC`	The PC to be plotted on the y axis.
`yr`	The y-xis range, in the format c(0,1)
`cols`	A vector of colors of length n, for use in scatterplot.
`title`	To be used for the plot
`png_dir`	A file path to a directory in which to save out PNG figures. Names will be automatically assigned based on input PC(s).

Details

Meant to be a quick diagnostic plot with minimal customization.

Value

Produces a series of plots to visualize PCA analysis. If `png_dir` is specified, function will save out `.png` files. Otherwise plots will be displayed in the default plot window.

`moRphomenses` *Geometric Morphometric Analysis of Hormone Cycle Phenotypes*

Description

Analyze shapes/phenotypes of hormone data using Geometric Morphometric inspired methods.

Author(s)

Daniel E. Ehrlich

`print_summary` *Print basic summary*

Description

Print basic summary

Usage

```
print_summary(aln, grps = NULL)
```

Arguments

`aln`	An object created with mm_ArrayData
`grps`	(Optional) A numeric vector that defines groupings

Value

A character vector with basic descriptive information, to be used with `print()`. If `grps=TRUE`, will return a list of character vectors.

Index