

# Package ‘prWarp’

July 23, 2025

**Title** Warping Landmark Configurations

**Version** 1.0.1

**Date** 2024-03-19

**Description** Compute bending energies, principal warps, partial warp scores, and the non-affine component of shape variation for 2D landmark configurations, as well as Mardia-Dryden distributions and self-similar distributions of landmarks, as described in Mitteroecker et al. (2020) <[doi:10.1093/sysbio/syaa007](https://doi.org/10.1093/sysbio/syaa007)>. Working examples to decompose shape variation into small-scale and large-scale components, and to decompose the total shape variation into outline and residual shape components are provided. Two landmark datasets are provided, that quantify skull morphology in humans and papionin primates, respectively from Mitteroecker et al. (2020) <[doi:10.5061/dryad.j6q573n8s](https://doi.org/10.5061/dryad.j6q573n8s)> and Grunstra et al. (2020) <[doi:10.5061/dryad.zkh189373](https://doi.org/10.5061/dryad.zkh189373)>.

**Depends** R (>= 3.6.0)

**Imports** Morpho

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Suggests** knitr, rmarkdown, geomorph

**VignetteBuilder** knitr

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Anne Le Maitre [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-2690-7367>>),  
Silvester Bartsch [aut],  
Nicole Grunstra [aut],  
Philipp Mitteroecker [aut]

**Maintainer** Anne Le Maitre <[anne.le.maitre@univie.ac.at](mailto:anne.le.maitre@univie.ac.at)>

**Repository** CRAN

**Date/Publication** 2024-03-20 18:00:02 UTC

Contents

|                         |           |
|-------------------------|-----------|
| array.to.xxyy . . . . . | 2         |
| create.pw.be . . . . .  | 3         |
| HomoMidSag . . . . .    | 4         |
| md.distri . . . . .     | 5         |
| papionin . . . . .      | 6         |
| ssim.distri . . . . .   | 7         |
| tps.all . . . . .       | 8         |
| xxyy.to.array . . . . . | 9         |
| <b>Index</b>            | <b>11</b> |

---

|               |                                                           |
|---------------|-----------------------------------------------------------|
| array.to.xxyy | <i>Convert (p x k x n) data array into 2D data matrix</i> |
|---------------|-----------------------------------------------------------|

---

Description

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

Usage

array.to.xxyy(A)

Arguments

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

Value

Function returns a two-dimensional matrix of dimension (n x [p x k]), where rows represent specimens and columns represent variables. The p first columns correspond to X coordinates, etc.

See Also

[xxyy.to.array](#)

Examples

```
A <- array(rnorm(40), c(5, 2, 4)) # random 2D coordinates of 5 landmarks for 4 specimens
array.to.xxyy(A)
```

create.pw.be

*Principal warps***Description**

Computes the principal warps and the bending energy of a reference shape configuration, as well as the variance of the partial warps, the partial warp scores and the non-affine component of shape variation for 2D landmark coordinates (3D not implemented). Small-scale and large-scale components of shape variation can also be computed.

**Usage**

```
create.pw.be(A, M_ref, d = NULL)
```

**Arguments**

|       |                                                                                                                                                                                      |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A     | a $k \times 2 \times n$ array, where $k$ is the number of 2D landmarks, and $n$ is the sample size.                                                                                  |
| M_ref | a $k \times 2$ reference matrix (usually the sample mean shape), where $k$ is the number of 2D landmarks                                                                             |
| d     | (optional) an integer value comprised between 1 and $(k-3)$ to compute small-scale shape components (between 1 and $d$ ) and large-scale shape components (between $d+1$ and $k-3$ ) |

**Value**

A list containing the following named components:

|                   |                                                                                                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bendingEnergy     | the bending energy (the $(k-3)$ eigenvalues of the bending energy matrix)                                                                                                     |
| principalWarps    | the $k \times (k-3)$ matrix of principal warps (the $k$ eigenvectors of the bending energy matrix)                                                                            |
| partialWarpScores | the $n \times (2k-6)$ matrix of partial warp (the projection of the vectors of shape coordinates, expressed as deviations from the reference shape, onto the principal warps) |
| variancePW        | the variance of the $(k-3)$ partial warps                                                                                                                                     |
| Xnonaf            | the $n \times 2k$ matrix of the non-affine component of shape variation                                                                                                       |
| Xsmall            | the $n \times 2k$ matrix of the small-scale shape variation (if $d$ is provided)                                                                                              |
| Xlarge            | the $n \times 2k$ matrix of the large-scale shape variation (if $d$ is provided)                                                                                              |

**References**

Bookstein FL. (1989). Principal Warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence* 11(6): 567–585. <https://ieeexplore.ieee.org/abstract/document/24792>

**See Also**

See [Createl](#) for the creation of the bending energy matrix

**Examples**

```
# 2D landmark coordinates
library("geomorph")
data("HomoMidSag") # dataset
n_spec <- dim(HomoMidSag)[1] # number of specimens
k <- dim(HomoMidSag)[2] / 2 # number of landmarks
homo_ar <- arrayspecs(HomoMidSag, k, 2) # create an array

# Procrustes registration
homo_gpa <- Morpho::procSym(homo_ar)
m_overall <- homo_gpa$rotated # Procrustes coordinates
m_mshape <- homo_gpa$mshape # average shape

# Computation of bending energy, partial warp scores etc.
homo_be_pw <- create.pw.be(m_overall, m_mshape)

# Partial warp variance as a function of bending energy
logInvBE <- log((homo_be_pw$bendingEnergy)^(-1)) # inverse log bending energy
logPWvar <- log(homo_be_pw$variancePW) # log variance of partial warps
mod <- lm(logPWvar ~ logInvBE) # linear regression
# Plot log PW variance on log BE^-1 with regression line
plot(logInvBE, logPWvar, col = "white", asp = 1,
main = "PW variance against inverse BE", xlab = "log 1/BE", ylab = "log PW variance")
text(logInvBE, logPWvar, labels = names(logPWvar), cex = 0.5)
abline(mod, col = "blue")
```

---

HomoMidSag

*HomoMidSag dataset*


---

**Description**

2D Cartesian coordinates of 87 landmarks quantifying the skull morphology along the midsagittal plane for 24 adult modern humans.

**Usage**

```
data(HomoMidSag)
```

**Format**

A data frame with 24 rows and 184 variables

## References

- Bartsch, Silvester (2019) The ontogeny of hominid cranial form: A geometric morphometric analysis of coordinated and compensatory processes. Master's thesis, University of Vienna.
- Mitteroecker, Philipp et al. (2020) Morphometric variation at different spatial scales: coordination and compensation in the emergence of organismal form. *Systematic Biology*, 69(5): 913–926. doi: 10.1093/sysbio/syaa007
- Mitteroecker, Philipp et al. (2020) Data form: Morphometric variation at different spatial scales: coordination and compensation in the emergence of organismal form. *Dryad Digital Repository*. doi: 10.5061/dryad.j6q573n8s

---

|           |                                   |
|-----------|-----------------------------------|
| md.distri | <i>Mardia-Dryden distribution</i> |
|-----------|-----------------------------------|

---

## Description

Create a matrix of 2D shape coordinates drawn from a Mardia-Dryden distribution (3D not implemented)

## Usage

```
md.distri(M_ref, n, sd = 0.05)
```

## Arguments

|       |                                                                                               |
|-------|-----------------------------------------------------------------------------------------------|
| M_ref | a k x 2 refence matrix (usually the sample mean shape), where k is the number of 2D landmarks |
| n     | the number of observations                                                                    |
| sd    | the standard deviation of the distribution (default = 0.02)                                   |

## Value

the n x 2k matrix of shape coordinates drawn from a Mardia-Dryden distribution

## Examples

```
# 2D landmark coordinates
library("geomorph")
data("HomoMidSag") # dataset
n_spec <- dim(HomoMidSag)[1] # number of specimens
k <- dim(HomoMidSag)[2] / 2 # number of landmarks
homo_ar <- arrayspecs(HomoMidSag, k, 2) # create an array

# Procrustes registration
homo_gpa <- Morpho::procSym(homo_ar)
m_mshape <- homo_gpa$mshape # average shape

# Mardia-Dryden distribution
```

```
Xmd <- md.distri(m_mshape, n = n_spec, sd = 0.005)
# Visualization
plot(Xmd[, 1:k], Xmd[, (k+1):(2*k)], asp = 1, las = 1, cex = 0.5,
main = "Mardia-Dryden distribution", xlab = "X", ylab = "Y")
```

---

papionin

*papionin dataset*

---

## Description

2D Cartesian coordinates of 70 landmarks quantifying the skull morphology along the midsagittal plane for 67 adult modern primates (mostly papionins). The data correspond to a list with the 6 following elements:

## Usage

```
data(papionin)
```

## Format

A list of 6 elements.

## Details

- **coords** The 3D array of landmark coordinates
- **species** The vector of species names
- **semi\_lm** The vector of semilandmark numbers of the full dataset
- **curves** The list of curves for sliding semilandmarks of the full dataset
- **links** The matrix of links between landmarks for the full dataset
- **outline** A list of 4 elements for the analysis of the outline shape: **subset**, the landmark numbers for the subset; **semi\_lm**, the vector of semilandmark numbers; **curves**, the list of curves for sliding semilandmarks; **links**, the matrix of links between landmarks for the full dataset.

## References

- Grunstra, Nicole D. S. et al. (2021) Detecting phylogenetic signal and adaptation in papionin cranial shape by decomposing variation at different spatial scales. *Systematic Biology*, 70(4): 694–706. doi: 10.1093/sysbio/syaa093
- Grunstra, Nicole D. S. et al. (2020) Data form: Detecting phylogenetic signal and adaptation in papionin cranial shape by decomposing variation at different spatial scales. *Dryad Digital Repository*. doi: 10.5061/dryad.zkh189373

---

|             |                                  |
|-------------|----------------------------------|
| ssim.distri | <i>Self-similar distribution</i> |
|-------------|----------------------------------|

---

## Description

Create a matrix of 2D shape coordinates drawn from a self-similar distribution (3D not implemented)

## Usage

```
ssim.distri(M_ref, n, sd = 0.02, f = 1)
```

## Arguments

|       |                                                                                                 |
|-------|-------------------------------------------------------------------------------------------------|
| M_ref | a k x 2 reference matrix (usually the sample mean shape), where k is the number of 2D landmarks |
| n     | the number of observations                                                                      |
| sd    | the standard deviation of the distribution (default = 0.02)                                     |
| f     | a scaling factor                                                                                |

## Value

the n x 2k matrix of shape coordinates drawn from a self-similar distribution

## Examples

```
# 2D landmark coordinates
library("geomorph")
data("HomoMidSag") # dataset
n_spec <- dim(HomoMidSag)[1] # number of specimens
k <- dim(HomoMidSag)[2] / 2 # number of landmarks
homo_ar <- arrayspecs(HomoMidSag, k, 2) # create an array

# Procrustes registration
homo_gpa <- Morpho::procSym(homo_ar)
m_mshape <- homo_gpa$mshape # average shape

# Self-similar distribution
Xdefl <- ssim.distri(m_mshape, n = n_spec, sd = 0.05, f = 1)
# Visualization
plot(Xdefl[, 1:k], Xdefl[, (k+1):(2*k)], asp = 1, las = 1, cex = 0.5,
main = "Self-similar distribution", xlab = "X", ylab = "Y")
```

---

|         |                                                                                       |
|---------|---------------------------------------------------------------------------------------|
| tps.all | <i>Thin plate spline mapping (2D and 3D) for several sets of landmark coordinates</i> |
|---------|---------------------------------------------------------------------------------------|

---

## Description

Maps landmarks via thin plate spline based on a reference and a target configuration in 2D or in 3D. This function is an extension of the tps3d function for a set of specimens.

## Usage

```
tps.all(X_array, REF_array, TAR_matrix)
```

## Arguments

|            |                                                                                                                                                             |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| X_array    | original coordinates - a 3D array (p x k x n) containing original landmark coordinates for a set of specimens                                               |
| REF_array  | reference coordinates (e.g., outline landmarks for all specimens) - a 3D array (p x k x n) containing reference landmark coordinates for a set of specimens |
| TAR_matrix | target coordinates (e.g., average outline landmarks) - a matrix (n x k) containing target landmark coordinates                                              |

## Details

p is the number of landmark points, k is the number of landmark dimensions (2 or 3), and n is the number of specimens.

## Value

Function returns a 3D array (p x k x n) containing the deformed input (original landmark set warped onto the target matrix).

## References

Bookstein FL. (1989). Principal Warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6): 567–585. <https://ieeexplore.ieee.org/abstract/document/24792>

## See Also

See [tps3d](#)

## Examples

```
data("papionin") # load dataset
# Full dataset: 70 landmarks
papionin_ar <- papionin$coords
# Outline dataset: subset of 54 landmarks
outline_ar <- papionin_ar[papionin$outline$subset, , ]

# Subset: Macaca only
mac <- grep("Macaca", papionin$species) # genus Macaca
papionin_macaca <- papionin_ar[, , mac]
outline_macaca <- outline_ar[, , mac]

# Landmark sliding by minimizing bending energy + superimposition (GPA)
library("Morpho")
papionin_gpa <- procSym(papionin_macaca, SMvector = papionin$semi_lm,
  outlines = papionin$curves)
outline_gpa <- procSym(outline_macaca, SMvector = papionin$outline$semi_lm,
  outlines = papionin$outline$curves)

# Warping the slid landmarks of the full landmark dataset to the average outline shape
residual_shape <- tps.all(X_array = papionin_gpa$dataslide,
  REF_array = outline_gpa$dataslide,
  TAR_matrix = outline_gpa$mshape)
```

---

xxyy.to.array

---

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


---

## Description

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

## Usage

```
xxyy.to.array(M, p, k = 2)
```

## Arguments

|   |                                                                                                                                                                                                                                                   |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| M | A matrix of dimension $n \times [p \times k]$ containing landmark coordinates for a set of specimens. Each row contains all landmark coordinates for a single specimen. The first columns correspond to the X coordinates for all landmarks, etc. |
| p | Number of landmarks                                                                                                                                                                                                                               |
| k | Number of dimensions (2 or 3)                                                                                                                                                                                                                     |

## Value

Function returns a 3D array ( $p \times k \times n$ ), 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.

**See Also**[array.to.xyyy](#)**Examples**

```
X <- matrix(rnorm(40), nrow = 4) # Random 2D coordinates of 5 landmarks for 4 specimens
xyyy.to.array(X, 5, 2)
```

# Index

`array.to.xxyy`, [2](#), [10](#)

`create.pw.be`, [3](#)

`CreateL`, [4](#)

`HomoMidSag`, [4](#)

`md.distri`, [5](#)

`papionin`, [6](#)

`ssim.distri`, [7](#)

`tps.all`, [8](#)

`tps3d`, [8](#)

`xxyy.to.array`, [2](#), [9](#)