

Package ‘morphomap’

May 9, 2026

Type Package

Title Morphometric Maps, Bone Landmarking and Cross Sectional Geometry

Version 1.5

Description Extract cross sections from long bone meshes at specified intervals along the diaphysis. Calculate two and three-dimensional morphometric maps, cross-sectional geometric parameters, and semilandmarks on the periosteal and endosteal contours of each cross section.

Depends R ($\geq 3.5.0$)

Imports Arothron (≥ 1.0), lattice (≥ 0.2), mgcv (≥ 1.8), Rvcg (≥ 0.18), Morpho (≥ 2.0), oce (≥ 1.1), sp (≥ 1.3), geometry ($\geq 0.4.0$), rgl (≥ 0.1), colorRamps (≥ 2.3), DescTools (≥ 0.99), grDevices (≥ 3.5), graphics (≥ 3.5)

License GPL-2

Encoding UTF-8

LazyLoad yes

RoxygenNote 7.2.3

NeedsCompilation no

Author Antonio Profico [aut, cre],
Luca Bondioli [aut],
Pasquale Raia [aut],
Julien Claude [ctb],
Paul O'Higgins [aut],
Damiano Marchi [aut]

Maintainer Antonio Profico <antonio.profico@gmail.com>

Repository CRAN

Date/Publication 2023-10-09 18:20:02 UTC

Contents

morphomap-package	2
Ex_mpShapeList	3
HomFem38023	3

morphomap2Dmap	3
morphomap3Dmap	6
morphomapAlignment	8
morphomapArea	9
morphomapArray2matrix	10
morphomapCentroid	10
morphomapCheck	11
morphomapCircle	12
morphomapCore	12
morphomapCSG	14
morphomapDF	17
morphomapExport	19
morphomapFlip	19
morphomapImport	20
morphomapMatrix2array	21
morphomapMirror	22
morphomapMoment	22
morphomapPCA	23
morphomapPic	24
morphomapPlotShape	26
morphomapReadMorphologika	28
morphomapRectangle	28
morphomapRegradius	29
morphomapSegm	30
morphomapShape	31
morphomapSort	33
morphomapThickness	34
morphomapTranslate	35
morphomapTri2sects	36
morphomapTriangulate	36
morphomapVariations	37
morphomapWriteMorphologika	38
morphomapZmoment	38
PanFem27713	39

Index **40**

morphomap-package *2D and 3D cortical thickness maps and cross sectional geometry*

Description

Tool to process long bone meshes (shape data, morphometric maps and cross-sectional geometry)

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O’Higgins, Damiano Marchi

Ex_mpShapeList *example dataset*

Description

morphomapShape objects from 5 femora

Usage

data(Ex_mpShapeList)

Author(s)

Antonio Profico

HomFem38023 *example dataset*

Description

3D mesh of a human femur bone

Usage

data(HomFem38023)

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

morphomap2Dmap *morphomap2Dmap*

Description

Create a 2D cortical thickness map

Usage

```

morphomap2Dmap(
  morphomap.shape,
  rem.out = FALSE,
  fac.out = 0.5,
  smooth = FALSE,
  scale = TRUE,
  smooth.iter = 5,
  gamMap = FALSE,
  nrow = 90,
  ncol = 100,
  gdl = 250,
  method = "equiangular",
  unwrap = "A",
  plot = TRUE,
  pal = blue2green2red(101),
  aspect = 2
)

```

Arguments

<code>morphomap.shape</code>	list: output from <code>morphomapShape</code> function
<code>rem.out</code>	logical: if TRUE the outlier will be removed
<code>fac.out</code>	numeric: parameter to set the threshold in outliers detection
<code>smooth</code>	logical: if TRUE a smooth filter is applied
<code>scale</code>	logical: if TRUE the thickness matrix is scaled from 0 to 1
<code>smooth.iter</code>	numeric: number of smoothing iterations
<code>gamMap</code>	logical: if TRUE gam smoothing is applied
<code>nrow</code>	numeric: number of rows for gam smoothing
<code>ncol</code>	numeric: number of columns for gam smoothing
<code>gdl</code>	numeric: number of degree of freedom for gam smoothing
<code>method</code>	character: if set on "equiangular" the cortical thickness is meant as the distance of the segment intersecting the external and internal outline starting from the centroid of the section. If set on "closest" the cortical thickness is calculated at each point as the closest distance between external and internal outlines
<code>unwrap</code>	character: starting quadrant to unwrap the diaphysis ("A"=anterior, "L"=lateral, "P"=posterior, "M"=mesial)
<code>plot</code>	logical: if TRUE the 2D morphometric map is plotted
<code>pal</code>	character vector: colors to be used in the map production
<code>aspect</code>	numeric: axis ratio for 2D morphometric map

Value

dataframe dataframe for colormap production
 2Dmap thickness color map
 gamoutput output from GAM
 data input used to build the GAM map

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```

if (interactive()){
library(colorRamps)
#morphomap on a human femur bone
data(HomFem38023)
meshes<-morphomapSegm(HomFem38023, param1=4)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-380.23
rawSections<-morphomapCore(out.sur=perMesh,
inn.sur=endMesh,num.sect=61,mech.len = mech_length, start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,24,sects_vector=NULL,cent.out="CCA",
delta=0.1, side="left")

#built 2D morphometric map without GAM smoothing
bone2Dmap<-morphomap2Dmap(morphomap.shape=shapeSections,
plot = TRUE, rem.out = TRUE,fac.out = 1.0, pal = blue2green2red(101),
aspect=2)
#built 2D morphometric map with GAM smoothing
bone2Dmap<-morphomap2Dmap(morphomap.shape=shapeSections,gam=TRUE,
plot = TRUE, rem.out = TRUE,fac.out = 1.0, pal = blue2green2red(101),
aspect=2)

#morphomap on a chimpanzee femur bone
data(PanFem27713)
meshes<-morphomapSegm(PanFem27713, param1=3)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-277.13
rawSections<-morphomapCore(out.sur=perMesh,
inn.sur=endMesh,num.sect=61,mech.len = mech_length, start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,24,sects_vector=NULL,cent.out="CCA",
delta=0.1, side="left")
#built 2D morphometric map without GAM smoothing
bone2Dmap<-morphomap2Dmap(morphomap.shape=shapeSections,plot = TRUE,
rem.out = TRUE,fac.out = 1.0,pal = blue2green2red(101),aspect=2)
#built 2D morphometric map with GAM smoothing
bone2Dmap<-morphomap2Dmap(morphomap.shape=shapeSections,gam=TRUE,
plot = TRUE, rem.out = TRUE,fac.out = 1.0,pal = blue2green2red(101),
aspect=2)

```

```
}

```

```
morphomap3Dmap      morphomap3Dmap
```

Description

Plot a 3D thickness map in four different anatomical views

Usage

```
morphomap3Dmap(
  morphomap.shape,
  out.sur,
  method = "equiangular",
  scale = TRUE,
  rem.out = FALSE,
  fac.out = 0.5,
  smooth = FALSE,
  smooth.iter = 5,
  k = 5,
  plot = TRUE,
  pal = blue2green2red(101)
)
```

Arguments

<code>morphomap.shape</code>	list: output from <code>morphomapShape</code> function
<code>out.sur</code>	3D mesh: 3D mesh of the long bone
<code>method</code>	character: if set on "equiangular" the cortical thickness is meant as the distance of the segment intersecting the external and internal outline starting from the centroid of the section. If set on "closest" the cortical thickness is calculated at each point as the closest distance between external and internal outlines
<code>scale</code>	logical: if TRUE the cortical thickness matrix will be scaled from 0 to 1
<code>rem.out</code>	logical: if TRUE outliers are identified and removed from thickness matrix
<code>fac.out</code>	numeric: parameter to set the threshold in outliers detection
<code>smooth</code>	logical: if TRUE the smoothing filter is applied on the thickness matrix
<code>smooth.iter</code>	numeric: number of smoothing iterations
<code>k</code>	integer: neighbourhood of kd-tree to search the nearest semilandmarks to each vertex
<code>plot</code>	logical: if TRUE the 3D map is plotted
<code>pal</code>	character vector: colors to be used in the map production

Value

cols color associated at each vertex of 3D mesh
 thickmat thickness matrix after smoothing and outliers removal

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```

if(interactive()){
#morphomap on a human femur bone
data(HomFem38023)
meshes<-morphomapSegm(HomFem38023, param1=4)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-380.23
rawSections<-morphomapCore(out.sur=perMesh,
inn.sur=endMesh,num.sect=61,
mech.len = mech_length,param1 = 0.5,
radius.fact = 2.5,npovs = 100,clean_int_out = TRUE,
num.points = 500, start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,24,sects_vector=NULL,cent.out="CCA",
delta=0.1, side="left")

#built 3D morphometric map
bone3Dmap<-morphomap3Dmap(shapeSections, out.sur=perMesh,
                           plot = TRUE,rem.out=TRUE,
                           fac.out=1.5,smooth=TRUE,
                           smooth.iter=5)

#or
require(rgl)
rgl::open3d()
rgl::shade3d(perMesh,col=bone3Dmap$cols,specular="black")

#morphomap on a chimpanzee femur bone
data(PanFem27713)
meshes<-morphomapSegm(PanFem27713, param1=3)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-277.13
rawSections<-morphomapCore(out.sur=perMesh,
                           inn.sur=endMesh,num.sect=61,mech.len = mech_length,
                           start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,24,sects_vector=NULL,cent.out="CCA",
delta=0.1, side="left")
#built 3D morphometric map
bone3Dmap<-morphomap3Dmap(shapeSections, out.sur=perMesh,
                           plot = TRUE,rem.out=TRUE,
                           fac.out=1.5,smooth=TRUE,
                           smooth.iter=5)

#or

```

```
require(rgl)
rgl::open3d()
rgl::shade3d(perMesh,col=bone3Dmap$cols,specular="black")
}
```

morphomapAlignment *morphomapAlignment*

Description

Align a femur bone following the protocol proposed by Ruff (2002)

Usage

```
morphomapAlignment(
  mesh,
  set,
  side = c("left", "right"),
  param1 = 4,
  iter1 = 2000,
  iter2 = 2000,
  iter3 = 2000,
  from1 = 180,
  to1 = 360,
  from2 = -5,
  to2 = 5,
  from3 = -5,
  to3 = 5,
  tol = 0.5
)
```

Arguments

mesh	3D mesh: femur long bone mesh
set	matrix: 7 landmarks acquired on the mesh (see details)
side	character: specify if the femur bone is "left" or "right" side
param1	numeric: parameter for spherical flipping (usually ranged between 3 and 4)
iter1	numeric: number of iterations first alignment
iter2	numeric: number of iterations second alignment
iter3	numeric: number of iterations third alignment
from1	numeric: inferior range of the allowed rotation in the first alignment
to1	numeric: superior range of the allowed rotation in the first alignment
from2	numeric: inferior range of the allowed rotation in the second alignment
to2	numeric: superior range of the allowed rotation in the second alignment
from3	numeric: inferior range of the allowed rotation in the third alignment
to3	numeric: superior range of the allowed rotation in the third alignment
tol	numeric: maximum allowed error in the alignment expressed in mm

Details

The function 'morphomapAlignment' is designed to align a femur bone. I did not tested on other long bones. The function requires 7 anatomical landmarks samples as follow: 1-the point at the center of the diaphysis in posterior view after the less trochanter, 2- the most posterior point on the lateral epicondyle, 3-the most posterior point on the medial epicondyle, 4- the most inferior point on the intercondilar fossa, 5- neck of the femur, 6- the most inferior point on the medial epicondyle and 7-the most inferior point on the lateral epicondyle. If the function in a short time does not complete the alignment, please stop the R session, check your landmark configuration or try to increase the value of the argument 'tol'.

Value

sur: mesh of the aligned femur bone

coo: coordinates of the landmark used in the alignment (plus two added automatically)

mech_length: mechanical length of the aligned femur bone

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

morphomapArea

morphomapArea

Description

Shoelace formula to calculate the area of a closed outline

Usage

```
morphomapArea(p, delta = 0.1, method = "shoelace")
```

Arguments

p matrix: kx2 matrix

delta numeric: picture elements of adjustable side length

method character: the user can choice to calculate the area applying the "shoelace" formula or discretizing the cross sections in dA areas (method = "delta")

Value

ar numeric: area

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```

extsec<-morphomapCircle(10,100)
#shoelace method
area<-morphomapArea(extsec, method="shoelace")
#delta method
area<-morphomapArea(extsec, method="delta",delta=0.01)

```

morphomapArray2matrix *morphomapArray2matrix*

Description

Convert an array into a matrix

Usage

```
morphomapArray2matrix(array)
```

Arguments

array an array

Value

mat a matrix

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

morphomapCentroid *morphomapCentroid*

Description

Calculate the barycenter of the cortical area

Usage

```
morphomapCentroid(cp, mp, delta = 0.1)
```

Arguments

cp matrix: coordinates of the external outline of the section
mp matrix: coordinates of the internal outline of the section
delta numeric: picture elements of adjustable side length

Value

centroid numeric vector: coordinates of the cortical area

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O’Higgins, Damiano Marchi

Examples

```
extsec<-morphomapCircle(10,100)
intsec<-morphomapCircle(8,100)
plot(extsec,asp=1,type="l")
points(intsec,col=2,type="l")
cent<-morphomapCentroid(extsec,intsec,delta = 0.1)
points(cent[1],cent[2],pch=19,col=3)
```

`morphomapCheck` *morphomapCheck*

Description

Plot the long bone mesh to check the orientation of the long bone

Usage

```
morphomapCheck(mesh, col = "white")
```

Arguments

<code>mesh</code>	3D mesh: long bone 3D model
<code>col</code>	character: color mesh

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O’Higgins, Damiano Marchi

Examples

```
library(morphomap)
data(HomFem38023)
morphomapCheck(HomFem38023)
```

morphomapCircle *morphomapCircle*

Description

Define a circular outline

Usage

```
morphomapCircle(r = 1, n = 1000)
```

Arguments

r numeric: radius of the outline
n numeric: number of points along the outline

Value

mat matrix with coordinates

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```
extsec<-morphomapCircle(10,100)  
intsec<-morphomapCircle(8,100)  
plot(extsec,asp=1,type="l")  
points(intsec,type="l",col=2)
```

morphomapCore *morphomapCore*

Description

Tool to build 3D and 2D cross sections

Usage

```

morphomapCore(
  out.sur = out.sur,
  inn.sur = inn.sur,
  num.sect = 61,
  mech.len,
  clean_int_out = TRUE,
  param1 = 0.5,
  radius.fact = 2.5,
  npovs = 100,
  num.points = 500,
  start = 0.2,
  end = 0.8,
  print.progress = TRUE
)

```

Arguments

out.sur	object of class mesh3d
inn.sur	object of class mesh3d
num.sect	number of sections
mech.len	mechanical length of the long bone
clean_int_out	logical if TRUE the inner section will be cleaned by using spherical flipping
param1	numeric parameter for spherical flipping operator (how much the section will be deformed)
radius.fact	numeric parameter for spherical flipping operator (distance from the center of the outline at which the povs are defined)
npovs	numeric: number of points of view defined around the section
num.points	number of equiangular points to be defined on each section
start	percentage of the mechanical length from which the first section is defined
end	percentage of the mechanical length from which the last section is defined
print.progress	logical: if TRUE a progress bar is printed to the screen

Value

3D_out num.pointsx3xnum.sect array of the external outlines
 3D_inn num.pointsx3xnum.sect array of the internal outlines
 2D_out num.pointsx2xnum.sect array of the external outlines
 2D_inn num.pointsx2xnum.sect array of the internal outlines
 mech_length mechanical length of the long bone
 start percentage of the mechanical length from which the first section is defined
 end percentage of the mechanical length from which the last section is defined

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```

if(interactive()){
#raw section on a human femur bone
data(HomFem38023)
meshes<-morphomapSegm(HomFem38023, param1=4)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-380.23
rawSections<-morphomapCore(out.sur=perMesh,
                           inn.sur=endMesh,num.sect=61,mech.len = mech_length,
                           start = 0.2,end=0.8)

#2D plot of the first section
plot(rawSections$`2D_out`[, ,1],col="grey",asp=1,xlab="x",ylab="y",type="l")
points(rawSections$`2D_inn`[, ,1],col="red",type="l")
#3D plot of the first section
require(rgl)
rgl::open3d()
rgl::plot3d(rawSections$`3D_out`[, ,1],aspect=FALSE,col="grey",
            type="l",lwd=5,xlab="x",ylab="y",zlab="z")
rgl::plot3d(rawSections$`3D_inn`[, ,1],aspect=FALSE,col="red",
            type="l",lwd=5,add=TRUE)

#raw section on a chimpanzee femur bone
data(PanFem27713)
meshes<-morphomapSegm(PanFem27713, param1=3)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-277.13
rawSections<-morphomapCore(out.sur=perMesh,
                           inn.sur=endMesh,num.sect=61,mech.len = mech_length,
                           start = 0.2,end=0.8)

#2D plot of the first section
plot(rawSections$`2D_out`[, ,1],col="grey",asp=1,xlab="x",ylab="y",type="l")
points(rawSections$`2D_inn`[, ,1],col="red",type="l")
#3D plot of the first section
require(rgl)
rgl::open3d()
rgl::plot3d(rawSections$`3D_out`[, ,1],aspect=FALSE,col="grey",
            type="l",lwd=5,xlab="x",ylab="y",zlab="z")
rgl::plot3d(rawSections$`3D_inn`[, ,1],aspect=FALSE,col="red",
            type="l",lwd=5,add=TRUE)
}

```

Description

Tool for Cross-sectional geometry

Usage

```

morphomapCSG(
  cp,
  mp,
  translate = FALSE,
  center = c("I", "E", "CCA"),
  delta = 0.1,
  Cx = NULL,
  Cy = NULL,
  I_xy = TRUE,
  I_minmax = TRUE,
  Zxy = TRUE
)

```

Arguments

cp	matrix: coordinates of the external outline
mp	matrix: coordinates of the internal outline
translate	logical: if TRUE the section will be centered
center	how to define the center of each section. The method allowed are "CCA" (center of cortical area), "E" (barycenter of the external outline) and "I" (barycenter of the internal outline)
delta	numeric: picture elements of adjustable side length
Cx	numeric: new x center coordinate
Cy	numeric: new y center coordinate
I_xy	logical: if TRUE the product of inertia around the x and y axis is calculated
I_minmax	logical: if TRUE the Imin and Imax will be calculated
Zxy	logical: if TRUE the polar moment of inertia will be calculated

Value

Cx x coordinate of the centered section
 Cy y coordinate of the centered section
 T_area total area
 M_area medullar area
 CA cortical area
 Ext_perim external perimeter
 Med_perim medullar perimiter
 Mean_thick mean thickness of the section

Sd_thick thickness standard deviation
 Min_thick minimum thickness
 Max_thick maximum thickness
 Ix numeric: moment of inertia around the x axis
 Iy numeric: moment of inertia around the y axis
 Zx numeric: moment of inertia around the x axis
 Zy numeric: moment of inertia around the y axis
 Zpol numeric: polar moment of inertia
 dx new centered coordinates of the internal outline
 dy new centered coordinates of the internal outline
 Imin numeric: minimum moment of inertia
 Imax numeric: maximum moment of inertia
 J numeric: polar moment of inertia
 Zmax numeric: the maximum polar section
 Zmin numeric: the minimum polar section
 theta numeric: theta angle

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```

#calculation of csg parameter on a human femur cross section
data(HomFem38023)
meshes<-morphomapSegm(HomFem38023, param1=4)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-380.23
rawSections<-morphomapCore(out.sur=perMesh,
                           inn.sur=endMesh,num.sect=61,mech.len = mech_length,
                           start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,250,sects_vector=NULL,cent.out="CCA",
delta=0.1, side="left")
csgSect31<-morphomapCSG(cp = shapeSections$`2D_out`[, ,31],
                       mp=shapeSections$`2D_inn`[, ,31],
                       translate = FALSE,center="CCA")

#Cross sectional geometry along the entire femur bone
results<-matrix(NA,ncol=24,nrow=61)
rownames(results)<-paste("section",c(1:61))
colnames(results)<-c("Cx","Cy","T_area","M_area","CA",
                   "Ext_perim","Med_perim","Mean_thick","Sd_thick" ,
                   "Min_thick","Max_thick","Ix","Iy","Zx" ,"Zy","Zpol" ,
                   "dx","dy","Imin","Imax","J","Zmax","Zmin","theta")

```

```

for(i in 1:61){
  results[i,]<-unlist(morphomapCSG(cp = shapeSections$`2D_out`[, ,i],
                                mp=shapeSections$`2D_inn`[, ,i],
                                translate = FALSE,center="CCA",delta = 0.5))
}

plot(c(1:61),results[,24],type="b",main="Theta",cex=1,
      xlab="section",ylab="radians")

#calculation of csg parameter on a chimpanzee femur cross section
data(PanFem27713)
meshes<-morphomapSegm(PanFem27713, param1=3)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-277.13
rawSections<-morphomapCore(out.sur=perMesh,
                           inn.sur=endMesh,num.sect=61,mech.len = mech_length,
                           start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,250,sects_vector=NULL,cent.out="CCA",
                              delta=0.1, side="left")
csgSect31<-morphomapCSG(cp = shapeSections$`2D_out`[, ,31],
                       mp=shapeSections$`2D_inn`[, ,31],
                       translate = FALSE,center="CCA")

#Cross sectional geometry along the entire femur bone
results<-matrix(NA,ncol=24,nrow=61)
rownames(results)<-paste("section",c(1:61))
colnames(results)<-c("Cx","Cy","T_area","M_area","CA",
                  "Ext_perim","Med_perim","Mean_thick","Sd_thick" ,
                  "Min_thick","Max_thick","Ix","Iy","Zx" ,"Zy","Zpol" ,
                  "dx","dy","Imin","Imax","J","Zmax","Zmin","theta")

for(i in 1:61){
  results[i,]<-unlist(morphomapCSG(cp = shapeSections$`2D_out`[, ,i],
                                mp=shapeSections$`2D_inn`[, ,i],
                                translate = FALSE,center="CCA",delta = 0.5))
}

plot(c(1:61),results[,24],type="b",main="Theta",cex=1,
      xlab="section",ylab="radians")

```

morphomapDF

morphomapDF

Description

Tool to build a data.frame suitable for morphometric maps

Usage

```

morphomapDF(
  morphomap.thickness,
  rem.out = TRUE,
  fac.out = 0.5,
  smooth = TRUE,
  scale = TRUE,
  smooth.iter = 5,
  method = "equiangular",
  unwrap = "A"
)

```

Arguments

morphomap.thickness	list: morphomap.Thickness object
rem.out	logical: if TRUE the outlier will be removed
fac.out	numeric: parameter to set the threshold in outliers detection
smooth	logical: if TRUE the smooth algorithm is applied
scale	logical: if TRUE the thichkness matrix is scaled from 0 to 1
smooth.iter	numeric: number of smoothing iterations
method	character: if set on "equiangular" the cortical thickness is meant as the distance of the segment intersecting the external and internal outline starting from the centroid of the section. If set on "closest" the cortical thickness is calculated at each point as the closest distance between external and internal outlines
unwrap	character: starting quadrant to unwrap the diaphysis ("A"=anterior, "L"=lateral, "P"=posterior, "M"=mesial)

Value

XYZ data.frame for morphometric map
 labels character vector for x labels in the morphometric map

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```

library(lattice)
library(colorRamps)
data(HomFem38023)
meshes<-morphomapSegm(HomFem38023, param1=4)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-380.23
rawSections<-morphomapCore(out.sur=perMesh,

```

```

inn.sur=endMesh,num.sect=61,mech.len = mech_length,
start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,21,sects_vector=NULL,cent.out="CCA",delta=0.1)
femthick<-morphomapThickness(shapeSections)
dataDF<-morphomapDF(femthick)$XYZ
contourplot(dataDF[, 3] ~ dataDF[, 1] + dataDF[, 2],
            col.regions=blue2green2red(101),region=TRUE,
            colorkey=list(at=seq(0,1,length.out = 100)),
            scales = list(x = list(at = seq(0,100,length.out = 5), c("A","M","P","L","A"),
alternating = 1)),asp=1.5,cuts=20,xlab="femur margin",ylab="biomechanical length")

```

morphomapExport

morphomapExport

Description

Export the output from ToothAlignement

Usage

```
morphomapExport(mpShapeObject, id, file)
```

Arguments

mpShapeObject	list: list containing morphomapShape objects
id	character: label name
file	character: name the output file

Author(s)

Antonio Profico

morphomapFlip

morphomapFlip

Description

Spherical flipping operator for bi-dimensional configuration

Usage

```
morphomapFlip(mat, param1 = 0.8, param2 = 10, radius.fact = 1.5, npovs = 100)
```

Arguments

mat	numeric matrix: coordinates of the bi-dimensional configuration
param1	numeric: first parameter for spherical flipping
param2	numeric: second parameter for spherical flipping
radius.fact	mechanical length of the long bone
npovs	number of evenly spaced points to be defined on each section

Value

mat matrix after spherical flipping

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```
#create a section
extsec<-morphomapCircle(10,1000)
intsec<-morphomapCircle(8,1000)
#simulate noise
noiseX<-rnorm(1000,mean = 0,sd = 0.2)
noiseY<-rnorm(1000,mean = 0,sd = 0.2)
noise<-cbind(noiseX,noiseY)
noisect<-intsec+noise
#spherical flipping
flipsect<-morphomapFlip(noisect,param1 = 2,radius.fact = 2)
sortsect<-morphomapSort(flipsect)
#original section
plot(extsec,asp=1,type="l",xlim=c(-15,15),ylim=c(-15,15))
points(intsec,asp=1,type="l",xlim=c(-15,15),ylim=c(-15,15))
#noise
points(noisect,col=2)
#new section after spherical flipping
points(sortsect,type="l",col=3,asp=1,lwd=2)
```

morphomapImport

morphomapImport

Description

Import a morphomapShape object exported with morphomapExport

Usage

morphomapImport(file)

Arguments

file character: name of input file

Value

3D_out num.pointsx3xnum.sect array in which the external outlines are stored
3D_inn num.pointsx3xnum.sect array in which the internal outlines are stored
2D_out num.pointsx2xnum.sect array in which the external outlines are stored
2D_inn num.pointsx2xnum.sect array in which the internal outlines are stored
ALPM_inn array with the coordinates of ALPM coordinates on the external outline
ALPM_out array with the coordinates of ALPM coordinates on the internal outline
mech_length mechanical length of the long bone
start percentage of the mechanical length from which the first section is defined
end percentage of the mechanical length from which the last section is defined

Author(s)

Antonio Profico

morphomapMatrix2array morphomapMatrix2array

Description

Convert a matrix into an array

Usage

`morphomapMatrix2array(matrix, nsects)`

Arguments

matrix an array
nsects number of cross sections

Value

array an array

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

morphomapMirror	<i>morphomapMirror</i>
-----------------	------------------------

Description

Mirror a long bone mesh along the yz plane

Usage

```
morphomapMirror(mesh)
```

Arguments

mesh	object of class mesh3d
------	------------------------

Value

mesh: object of class mesh3d

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```
if(interactive()){
#a left human femur bone
require(rgl)
data(HomFem38023)
lfem<-HomFem38023
rfem<-morphomapMirror(lfem)
rgl::open3d()
rgl::wire3d(lfem,col="green")
rgl::ire3d(rfem,col="red")
}
```

morphomapMoment	<i>morphomapMoment</i>
-----------------	------------------------

Description

Calculate the moment of inertia around the x and y axes and the product of inertia

Usage

```
morphomapMoment(cp, mp, delta = 0.1)
```

Arguments

cp matrix: coordinates of the external outline
 mp matrix: coordinates of the internal outline
 delta numeric: picture elements of adjustable side length

Value

Ix numeric: moment of inertia around the x axis
 Iy numeric: moment of inertia around the y axis
 Ixy numeric: product of inertia around the x and y axis

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```
#create a section
extsec<-morphomapCircle(10,1000)
intsec<-morphomapCircle(8,1000)
InMs<-morphomapMoment(extsec,intsec,delta=0.1)
```

morphomapPCA

morphomapPCA

Description

Calculate maps of cortical thickness and perform a Principal Component Analysis

Usage

```
morphomapPCA(
  mpShapeList,
  gamMap = TRUE,
  nrow = 61,
  ncol = 24,
  rem.out = TRUE,
  scaleThick = FALSE,
  fac.out = 1.5,
  method = "equiangular",
  scalePCA = TRUE,
  unwrap = "A"
)
```

Arguments

<code>mpShapeList</code>	list: list containing <code>morphomapShape</code> objects
<code>gamMap</code>	list: list containing <code>morphomapShape</code> objects
<code>nrow</code>	list: list containing <code>morphomapShape</code> objects
<code>ncol</code>	list: list containing <code>morphomapShape</code> objects
<code>rem.out</code>	list: list containing <code>morphomapShape</code> objects
<code>scaleThick</code>	list: list containing <code>morphomapShape</code> objects
<code>fac.out</code>	list: list containing <code>morphomapShape</code> objects
<code>method</code>	list: list containing <code>morphomapShape</code> objects
<code>scalePCA</code>	list: list containing <code>morphomapShape</code> objects
<code>unwrap</code>	list: list containing <code>morphomapShape</code> objects

Value

PCscores PC scores
 PCs loadings
 Variance Table of the explained Variance by the PCs
 meanMap mean morphometric map
 CorMaps morphometric maps

Author(s)

Antonio Profico

Examples

```
data(Ex_mpShapeList)
PCA<-morphomapPCA(Ex_mpShapeList)
plot(PCA$PCscores)
barplot(PCA$Variance[,2])
```

`morphomapPic`

morphomapPic

Description

Save the sections defined via `morphomapShape` or `morphomapCore`

Usage

```

morphomapPic(
  morphomap.core,
  morphomap.shape,
  vector = NULL,
  full = TRUE,
  width = 1500,
  height = 1500,
  pointsize = 12,
  res = 300,
  colthk = "red",
  collbs = "blue",
  dirpath = tempdir()
)

```

Arguments

morphomap.core	list: morphomap.core object
morphomap.shape	list: morphomap.shape object
vector	numeric: define which sections will be saved
full	logical: if TRUE the thickness at ALPM is reported
width	numeric: width of the picture
height	numeric: height of the picture
pointsize	numeric: pointsize of plotted text
res	numeric: the nominal resolution in ppi which will be recorded
colthk	specify the color for the numbers
collbs	specify the color for the labels
dirpath	character: path of the directory where the pictures will be saved

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```

#export picture from a human femur bone
data(HomFem38023)
meshes<-morphomapSegm(HomFem38023, param1=4)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-380.23
rawSections<-morphomapCore(out.sur=perMesh,
                           inn.sur=endMesh,num.sect=11,mech.len = mech_length,
                           start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,250,sects_vector=NULL,cent.out="CCA",

```

```

delta=0.5, side="left")
morphomapPic(rawSections,shapeSections,full=TRUE,dirpath=tempdir(),
             width=2500,height=2500)

#export picture from a chimpanzee femur bone
data(PanFem27713)
meshes<-morphomapSegm(PanFem27713, param1=3)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-277.13
rawSections<-morphomapCore(out.sur=perMesh,
                          inn.sur=endMesh,num.sect=11,mech.len = mech_length,
                          start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,250,sects_vector=NULL,cent.out="CCA",delta=0.5,
                              side="left")
morphomapPic(rawSections,shapeSections,full=TRUE,dirpath=tempdir(),
             width=2500,height=2500)

```

`morphomapPlotShape` *morphomapPlotShape*

Description

Visualize 2D and 3D cross sections

Usage

```

morphomapPlotShape(
  Shape,
  dims = 3,
  col1 = "red",
  col2 = "green",
  colc = "orange",
  colr = "violet",
  coll1 = "darkred",
  coll2 = "darkgreen",
  size = 1.5,
  lwd = 0.7,
  colmesh1 = "red",
  colmesh2 = "green",
  alpha = 0.3,
  tri = TRUE,
  outlines = TRUE,
  points = TRUE,
  lines = FALSE,
  centroid = FALSE,
  cent.out = "CCA",

```

```

    delta = 0.1,
    vecs = NULL
)

```

Arguments

Shape	list: output from morphomapShape function
dims	numeric: 2 = bi-dimensional cross sections, 3 = three-dimensional cross sections
col1	color of the external outline
col2	color of the internal outline
colc	color of the centroid of the cross section
colr	color of the radii
coll1	color of the lines on the external outline
coll2	color of the lines on the internal outline
size	numeric: points and spheres size
lwd	numeric: line width in pixels
colmesh1	color of the periosteal mesh
colmesh2	color of the endosteal mesh
alpha	numeric: alpha value between 0 (fully transparent) and 1 (opaque)
tri	logical: if TRUE the semilandmarks configuration is triangulated
outlines	logical: if TRUE the 2D and 3D outlines are plotted
points	logical: if TRUE points (2D) and spheres (3D) are plotted
lines	logical: if TRUE 2D and 3D lines are plotted
centroid	logical: if TRUE 2D and 3D centroids are plotted
cent.out	how to define the center of each section. The method allowed are "CCA" (center of cortical area), "E" (barycenter of the external outline) and "I" (barycenter of the internal outline)
delta	pixel size used to calculate the CCA
vecs	numeric: which sections will be plotted. If dims is set on 2 only the first element of the vector vecs is considered

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```

if(interactive()){
#morphomap on a human femur bone
data(HomFem38023)
meshes<-morphomapSegm(HomFem38023)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-380.23
}

```

```

rawSections<-morphomapCore(out.sur=perMesh,
inn.sur=endMesh,num.sect=61,mech.len = mech_length, start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,21,sects_vector=NULL,cent.out="CCA",
delta=0.1, side="left")
#Plot the object morphomapShape in three dimensions
morphomapPlotShape(shapeSections,dims=3, size=0.5)
#Plot a 2D cross-section
morphomapPlotShape(shapeSections,dims=2,lines=TRUE,vecs=31)
}

```

morphomapReadMorphologika

morphomapReadMorphologika

Description

Import an array stored in a morphologika file

Usage

```
morphomapReadMorphologika(file)
```

Arguments

file path of the file to be read

Value

out list containing an array, labels, groups and variables

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

morphomapRectangle

morphomapRectangle

Description

Define a rectangular outline

Usage

```
morphomapRectangle(l = 1, h = 1, n = 1000)
```

Arguments

l	numeric: length of the rectangle
h	numeric: height of the rectangle
n	numeric: number of points along the outline

Value

mat matrix with coordinates

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```
extsec<-morphomapRectangle(10,6,100)
intsec<-morphomapRectangle(8,4,100)
plot(extsec,asp=1,type="l")
points(intsec,type="l",col=2)
```

morphomapRegradius *morphomapRegradius*

Description

Wrapper of the function `regularradius` written by Julien Claude (Morphometrics with R)

Usage

```
morphomapRegradius(mat, center, n)
```

Arguments

mat	a kx2 matrix
center	coordinates of the center from which the calculation of regular radius started
n	number of points

Value

V2 position of landmarks equi angular spaced

Author(s)

Julien Claude, Antonio Profico

References

Claude, J. (2008). *Morphometrics with R*. Springer Science & Business Media.

Examples

```

extsec<-morphomapCircle(10,1000)
sel<-morphomapRegradius(extsec,center = c(0,0),n=11)
selcoo<-extsec[sel,]
plot(extsec,type="l",asp=1)
points(selcoo,col="red",pch=19)

```

morphomapSegm

morphomapSegm

Description

Separate a mesh from its visible and not visible components by using CA-LSE method

Usage

```

morphomapSegm(mesh, views = 30, param1 = 4, num.cores = NULL)

```

Arguments

mesh	object of class mesh3d
views	numeric: number of points of view
param1	numeric: first parameter for spherical flipping (usually ranged between 3 and 4)
num.cores	numeric: number of cores

Details

The result could be affected by the value set in the param1 argument. Before running morphomapCore please the periosteal and endosteal surfaces.

Value

external mesh3d of the visible facets from the points of view
internal mesh3d of the not visible facets from the points of view

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

References

Profico A., Schlager S., Valoriani V., Buzi C., Melchionna M., Veneziano A., Raia P., Moggi-Cecchi J. and Manzi G., 2018. Reproducing the internal and external anatomy of fossil bones: Two new automatic digital tools. American Journal of Physical Anthropology 166(4): 979-986.

Examples

```

if(interactive()){
#automatic separation of external and medullar femur components
require(rgl)
data(HomFem38023)
meshes<-morphomapSegm(HomFem38023,param1=4)
perMesh<-meshes$external
endMesh<-meshes$internal
rgl::open3d()
rgl::wire3d(perMesh,col="grey")
rgl::wire3d(endMesh,col="red")
}

```

morphomapShape

morphomapShape

Description

Tool for the extraction of equiangular landmarks on the entire diaphysis

Usage

```

morphomapShape(
  morphomap.core,
  num.land,
  sects_vector,
  cent.out = "CCA",
  delta = 0.1,
  side = "left"
)

```

Arguments

morphomap.core	list: morphomap.core object
num.land	numeric: number of landmarks defining each section
sects_vector	numeric: number of sections
cent.out	how to define the center of each section. The method allowed are "CCA" (center of cortical area), "E" (barycenter of the external outline) and "I" (barycenter of the internal outline)
delta	pixel size used to calculate the CCA
side	character: specify if the long bone is "left" or "right" side

Value

3D_out num.pointsx3xnum.sect array in which the external outlines are stored
 3D_inn num.pointsx3xnum.sect array in which the internal outlines are stored
 2D_out num.pointsx2xnum.sect array in which the external outlines are stored
 2D_inn num.pointsx2xnum.sect array in which the internal outlines are stored
 ALPM_inn array with the coordinates of ALPM coordinates on the external outline
 ALPM_out array with the coordinates of ALPM coordinates on the internal outline
 mech_length mechanical length of the long bone
 start percentage of the mechanical length from which the first section is defined
 end percentage of the mechanical length from which the last section is defined

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```
library(morphomap)
data(HomFem38023)
meshes<-morphomapSegm(HomFem38023, param1=4)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-380.23
rawSections<-morphomapCore(out.sur=perMesh, inn.sur=endMesh, num.sect=61 ,
mech.len = mech_length, start = 0.2,end=0.8,num.points = 500)
# Shape coordinates defining as center the barycenter of the cortical area
shapeSections_CCA<-morphomapShape(rawSections,21,sects_vector=NULL, cent.out="CCA",
delta=0.1,side="left")
# First the first cross section (2D)
morphomapPlotShape(shapeSections_CCA,dims=2,cent.out="CCA",vecs=1)
# First the first cross section (3D)
morphomapPlotShape(shapeSections_CCA,dims=3,size=0.5,lwd=2,cent.out="I",vecs=1)
# The entire diaphysis (3D)
morphomapPlotShape(shapeSections_CCA,dims=3,size=0.5,lwd=2,cent.out="I",vecs=NULL)
# Shape coordinates defining as center the barycenter of the external perimeter
shapeSections_E<-morphomapShape(rawSections, 21, sects_vector=NULL, cent.out="E",
delta=0.1, side="left")
# First the first cross section (2D)
morphomapPlotShape(shapeSections_E,dims=2,cent.out="E",vecs=1)
# First the first cross section (3D)
morphomapPlotShape(shapeSections_E,dims=3,size=0.5,lwd=2,cent.out="I",vecs=1)
# The entire diaphysis (3D)
morphomapPlotShape(shapeSections_E,dims=3,size=0.5,lwd=2,cent.out="I",vecs=NULL)

# Shape coordinates defining as center the barycenter of the internal perimeter
shapeSections_I<-morphomapShape(rawSections, 21, sects_vector=NULL, cent.out="I",
delta=0.1, side="left")
# First the first cross section (2D)
```

```
morphomapPlotShape(shapeSections_I,dims=2,lines=TRUE,cent.out="I",vecs=1)
# First the first cross section (3D)
morphomapPlotShape(shapeSections_I,dims=3,lines=TRUE,centroid=TRUE, size=0.5,
lwd=2,cent.out="I",vecs=1)
# The entire diaphysis (3D)
morphomapPlotShape(shapeSections_I,dims=3,size=0.5,lwd=2,cent.out="I",vecs=NULL)
```

morphomapSort	<i>morphomapSort</i>
---------------	----------------------

Description

Sort a series of points stored as a 2D matrix

Usage

```
morphomapSort(mat)
```

Arguments

mat numeric matrix: a kx2 matrix

Value

mat sorted kx2 matrix

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```
rand<-sample(100)
extsec<-morphomapCircle(10,100)[rand,]
plot(extsec,type="l",asp=1)
sorted<-morphomapSort(extsec)
plot(sorted,type="l",asp=1)
```

morphomapThickness *morphomapThickness*

Description

Tool for the extraction of equiangular landmarks on the entire diaphysis

Usage

```
morphomapThickness(morphomap.shape)
```

Arguments

morphomap.shape
list: morphomap.shape object

Value

sect_thickness cortical thickness at each pair of landmarks on the external and internal outlines
ALPM_thickness cortical thickness at ALPM quadrants

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```
#morphomap on a human femur bone
data(HomFem38023)
meshes<-morphomapSegm(HomFem38023, param1=4)
perMesh<-meshes$external
endMesh<-meshes$internal
mech_length<-380.23
rawSections<-morphomapCore(out.sur=perMesh,
                           inn.sur=endMesh,num.sect=61,mech.len = mech_length,
                           start = 0.2,end=0.8)
shapeSections<-morphomapShape(rawSections,21,sects_vector=NULL,cent.out="CCA",delta=0.1)
femthick<-morphomapThickness(shapeSections)
plot(femthick$ALPM_thickness[1,,],type="l",
     main="LAMP thickness",xlab="section",ylab="thickness")
points(femthick$ALPM_thickness[2,,],type="l",col=2)
points(femthick$ALPM_thickness[3,,],type="l",col=3)
points(femthick$ALPM_thickness[4,,],type="l",col=4)
```

morphomapTranslate *morphomapTranslate*

Description

Translate a section to a new center defined by the user

Usage

```
morphomapTranslate(corA, medA, Cx, Cy)
```

Arguments

corA	matrix: coordinates of the external outline
medA	matrix: coordinates of the internal outline
Cx	numeric: new x center coordinate
Cy	numeric: new y center coordinate

Value

cortical new centered coordinates of the external outline
medullar new centered coordinates of the internal outline

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```
extsec<-morphomapCircle(10,1000)
intsec<-morphomapCircle(8,1000)
plot(extsec,asp=1,type="l",xlim=c(-11,11),ylim=c(-11,11))
points(intsec,type="l")
traSect<-morphomapTranslate(extsec,intsec,1,1)
points(traSect$cortical,type="l",col="red")
points(traSect$medullar,type="l",col="red")
```

morphomapTri2sects *morphomapTri2sects*

Description

Triangulate the external and internal outlines of a 3D cross section

Usage

morphomapTri2sects(cp, mp)

Arguments

cp	matrix: coordinates of the external outline of the section
mp	matrix: coordinates of the internal outline of the section

Value

matrix coordinates of the triangulated mesh
 tri triangulations of the triangulated mesh

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

morphomapTriangulate *morphomapTriangulate*

Description

Build a mesh starting from the coordinates of the diaphysis

Usage

morphomapTriangulate(set, n, close = FALSE)

Arguments

set	matrix: coordinates of the cross sections to be triangulated
n	numeric: number of cross sections
close	logical: if TRUE the two surfaces are closed

Value

mesh a mesh of the triangulated semilandark configuration

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

morphomapVariations *morphomapVariations*

Description

Calculate cortical map variation from PCA

Usage

```
morphomapVariations(PCA, scores, PC, pal = blue2green2red(101), asp = 2)
```

Arguments

PCA	list: list containing morphomapShape objects
scores	list: list containing morphomapShape objects
PC	list: list containing morphomapShape objects
pal	list: list containing morphomapShape objects
asp	numeric: aspect ratio of the morphometric map

Value

mapvar: matrix containing values of cortical thickness

Author(s)

Antonio Profico

Examples

```
data(Ex_mpShapeList)
PCA<-morphomapPCA(Ex_mpShapeList)
plot(PCA$PCscores)
barplot(PCA$Variance[,2])
morphomapVariations(PCA,min(PCA$PCscores[,1]),PCA$PCs[,1])
morphomapVariations(PCA,max(PCA$PCscores[,1]),PCA$PCs[,1])
```

morphomapWriteMorphologika
morphomapWriteMorphologika

Description

Export an array in the morphologika format file

Usage

```
morphomapWriteMorphologika(array, groups = NULL, variables = NULL, file)
```

Arguments

array	an array
groups	a vector containing a classifier
variables	list containing further classifiers
file	path of the file to be saved

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

morphomapZmoment *morphomapZmoment*

Description

Calculate the polar moment of inertia around the x and y axes and the polar section module

Usage

```
morphomapZmoment(cp, mp, Cx = 0, Cy = 0, delta = 0.1)
```

Arguments

cp	matrix: coordinates of the external outline of the section
mp	matrix: coordinates of the internal outline of the section
Cx	numeric: x coordinate of the section center
Cy	numeric: y coordinate of the section center
delta	numeric: picture elements of adjustable side length

Value

Zx numeric: moment of inertia around the x axis

Zy numeric: moment of inertia around the y axis

dx numeric: maximum chord length from y axis

dy numeric: maximum chord length from x axis

Zpol numeric: polar moment of inertia

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Examples

```
extsec<-morphomapCircle(10,1000)
intsec<-morphomapCircle(8,1000)
ZMs<-morphomapZmoment(extsec,intsec,delta=0.1)
```

PanFem27713

example dataset

Description

3D mesh of a chimpanzee femur bone

Usage

```
data(PanFem27713)
```

Author(s)

Antonio Profico, Luca Bondioli, Pasquale Raia, Paul O'Higgins, Damiano Marchi

Index

* **morphomap**

- Ex_mpShapeList, [3](#)
- HomFem38023, [3](#)
- PanFem27713, [39](#)

Ex_mpShapeList, [3](#)

HomFem38023, [3](#)

morphomap (morphomap-package), [2](#)

morphomap-package, [2](#)

morphomap2Dmap, [3](#)

morphomap3Dmap, [6](#)

morphomapAlignment, [8](#)

morphomapArea, [9](#)

morphomapArray2matrix, [10](#)

morphomapCentroid, [10](#)

morphomapCheck, [11](#)

morphomapCircle, [12](#)

morphomapCore, [12](#)

morphomapCSG, [14](#)

morphomapDF, [17](#)

morphomapExport, [19](#)

morphomapFlip, [19](#)

morphomapImport, [20](#)

morphomapMatrix2array, [21](#)

morphomapMirror, [22](#)

morphomapMoment, [22](#)

morphomapPCA, [23](#)

morphomapPic, [24](#)

morphomapPlotShape, [26](#)

morphomapReadMorphologika, [28](#)

morphomapRectangle, [28](#)

morphomapRegradius, [29](#)

morphomapSegm, [30](#)

morphomapShape, [31](#)

morphomapSort, [33](#)

morphomapThickness, [34](#)

morphomapTranslate, [35](#)

morphomapTri2sects, [36](#)

morphomapTriangulate, [36](#)

morphomapVariations, [37](#)

morphomapWriteMorphologika, [38](#)

morphomapZmoment, [38](#)

PanFem27713, [39](#)