Package 'Toothnroll'

May 7, 2026

Title Dental Tissues Landmarking Measuring and Mapping

Version 1.11

Description Two- and three-dimensional morphometric maps of enamel and dentine thickness and multivariate analysis.
Volume calculation of dental materials.
Principal component analysis of thickness maps with associated morphometric map variations.

Imports Arothron (>= 1.0), alphashape3d (>= 1.0), compositions(>= 2.0), morphomap (>= 1.0), stringr (>= 1.0), vegan,lattice (>= 0.2), mgcv (>= 1.8), Rvcg (>= 0.18), Morpho (>= 2.0), oce (>= 1.1), rgl (>= 0.1), geometry (>= 0.4.0), colorRamps (>= 2.3), DescTools (>= 0.99), grDevices (>= 3.5), graphics (>= 3.5)

License GPL-2

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Antonio Profico [cre, aut],
Mathilde Augoyard [aut]

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

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2024-04-12 15:20:22 UTC

Contents

`Tooth2Dmap` *Tooth2Dmap*

Description

Create 2D morphometric maps of enamel/dentin thickness

Usage

```
Tooth2Dmap(
  tooth.shape,
  input,
  rem.out = FALSE,
  fac.out = 0.5,
  smooth = FALSE,
  scale = TRUE,
  smooth.iter = 5,
  gamMap = FALSE,
  nrow = 120,
  ncol = 80,
  gdl = 250,
  method = "equiangular",
  plot = TRUE,
  pal = blue2green2red(101),
  aspect = 0.6,
  labels = c("Li", "Mes", "Bu", "D", "Li"),
  ylab = ""
)
```

Arguments

tooth.shape	list: output from the function ToothShape
input	list: output from the function ToothAlignment
rem.out	logical: if TRUE outliers will be removed
fac.out	numeric: parameter to set the threshold in outlier detection
smooth	logical: if TRUE a smoothing filter is applied
scale	logical: if TRUE the thichkness matrix is scaled from 0 to 1
smooth.iter	numeric: number of smoothing iterations
gamMap	logical: if TRUE gam smoothing is applied
nrow	numeric: number of rows for gam smoothing matrix
ncol	numeric: number of columns for gam smoothing matrix
gdl	numeric: number of degree of freedom for gam smoothing matrix
method	character: if set on "equiangular" the dentine or enamel 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 dentine or enamel thickness is calculated at each point as the closest distance between external and internal outlines
plot	logical: if TRUE the 2D morphometric map is plotted
pal	character vector: colors to be used in the map production
aspect	numeric: axis ratio for 2D morphometric map
labels	character vector: names for x labels in the morphometric map
ylab	character vector: label for y axis in the 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; Mathilde Augoyard

Examples

```
data("URI1_tooth")
require(morphomap)
Enamel<-URI1_tooth$mesh1
Dentin<-URI1_tooth$mesh2
Pulp<-URI1_tooth$mesh3
outline<-URI1_tooth$outline
set<-URI1_tooth$set
```

```
#Map of the crown
AlignMeshes<-ToothAlignment(mesh1=Enamel,mesh2=Dentin,mesh3=Pulp,set,outline,analyse = "c")
#Virtual sectioning dentine-pulp
External<-AlignMeshes$almesh1$mesh
Internal<-AlignMeshes$almesh2$mesh
#Define 16 cross-sections from the 30% to the 90% along the crown
Core<-ToothCore(External,Internal,num.points = 1000,num.sect =16,
                    bio.len = AlignMeshes$length,start=0.3,end=0.9)
#Extract 25 equiangular semilandmarks from each cross-section (anticlockwise)
Shape<-ToothShape(Core,25,secta_vector = NULL,cent.out = "E",direction="a")
Tooth2Dmap(Shape,AlignMeshes,rem.out =TRUE,scale=FALSE,smooth = FALSE,aspect = 0.5,gamMap = FALSE,
           nrow = 100,ncol = 100,gdl = 250,method="equiangular")

#Map of the root
AlignMeshes<-ToothAlignment(mesh1=Enamel,mesh2=Dentin,mesh3=Pulp,set,outline,analyse = "r")
#Virtual sectioning dentine-pulp
External<-AlignMeshes$almesh2$mesh
Internal<-AlignMeshes$almesh3$mesh
#Define 16 cross-sections from the 10% to the 50% along the root
Core<-ToothCore(External,Internal,num.points = 1000,num.sect =16,
                    bio.len = AlignMeshes$length,start=0.1,end=0.5)
#Extract 25 equiangular semilandmarks from each cross-section (anticlockwise)
Shape<-ToothShape(Core,25,secta_vector = NULL,cent.out = "E",direction="a")
Tooth2Dmap(Shape,AlignMeshes,rem.out =FALSE,scale=FALSE,smooth = FALSE,aspect = 0.5,gamMap = FALSE,
           nrow = 100,ncol = 100,gdl = 250,method="equiangular")
```

ToothAlignment *ToothAlignment*

Description

Align dental meshes using as reference the cervical outline and five landmarks

Usage

```
ToothAlignment(mesh1, mesh2, mesh3 = NULL, set, outline, analyse = c("r", "c"))
```

Arguments

mesh1	3D mesh: dental mesh (enamel)
mesh2	3D mesh: dental mesh (dentin)
mesh3	3D mesh: dental mesh (dental pulp)
set	matrix: 5 landmarks acquired on the mesh (see details)
outline	matrix: set of coordinates along the cerical outline
analyse	character: "r" for root, "c" for crown

Details

The function needs five landmarks to align the dental meshes. Usually landmarks from 1 to 4 define the Lingual, Mesial, Buccal and Distal margins. The fifth landmark defines the end of the z axis (biomechanical length). The centroid of the cervical outline defines the origin of axes.

Value

almesh1: mesh of the aligned mesh1

almesh2: mesh of the aligned mesh2

almesh3: mesh of the aligned mesh3

alset: coordinates of the aligned landmark configuration

length: biomechanical length of the root (see details)

margins: coordinates of the landmarks in correspondence of the four margins

margins_sel: position of the margin along the outline

aloutline: coordinates of the aligned cervical outline

diamBL: bucco-lingual diameter

diamMD: mesio-distal diameter

Author(s)

Antonio Profico; Mathilde Augoyard

Examples

```
data("URI1_tooth")
Enamel<-URI1_tooth$mesh1
Dentin<-URI1_tooth$mesh2
Pulp<-URI1_tooth$mesh3
outline<-URI1_tooth$outline
set<-URI1_tooth$set

#Example on the root
AlignMeshesR<-ToothAlignment(mesh1=Enamel,mesh2=Dentin,mesh3=Pulp,set,outline,analyse = "r")
  require(rgl)
  open3d()
  shade3d(AlignMeshesR$almesh1$mesh,col="white",alpha=0.5)
  shade3d(AlignMeshesR$almesh2$mesh,col="pink",alpha=0.5,add=TRUE)
  shade3d(AlignMeshesR$almesh3$mesh,col="orange",alpha=0.5,add=TRUE)
  spheres3d(AlignMeshesR$alset,radius=0.25)
  spheres3d(AlignMeshesR$outline,radius=0.1,col="blue")
  lines3d(AlignMeshesR$outline)
  text3d(rbind(AlignMeshesR$outline[AlignMeshesR$margins_sel,],AlignMeshesR$alset[4,]),
  texts=1:5,cex=4)
  spheres3d(AlignMeshesR$alset[2,],radius=0.3,col="red")
  arrow3d(colMeans(AlignMeshesR$aloutline),AlignMeshesR$alset[4,],lwd=3,col="green",
  type="lines",s=1/10)
  lines3d(rbind(AlignMeshesR$alset[2,],AlignMeshesR$alset[1,]),lwd=3,col="green")
  axes3d()
```

```
#Example on the crown

AlignMeshesC<-ToothAlignment(mesh1=Enamel,mesh2=Dentin,mesh3=Pulp,set,outline,
analyse = "c")
  require(rgl)
  open3d()
  shade3d(AlignMeshesC$almesh1$mesh,col="white",alpha=0.5)
  shade3d(AlignMeshesC$almesh2$mesh,col="pink",alpha=0.5,add=TRUE)
  shade3d(AlignMeshesC$almesh3$mesh,col="orange",alpha=0.5,add=TRUE)
  spheres3d(AlignMeshesC$alset,radius=0.25)
  spheres3d(AlignMeshesC$outline,radius=0.1,col="blue")
  lines3d(AlignMeshesC$outline)
  text3d(rbind(AlignMeshesC$outline[AlignMeshesC$margins_sel,],AlignMeshesC$alset[4,]),
  texts=1:5,cex=4)
  spheres3d(AlignMeshesC$alset[2,],radius=0.3,col="red")
  arrow3d(colMeans(AlignMeshesC$aloutline),AlignMeshesC$alset[4,],lwd=3,col="green",
  type="lines",s=1/10)
  lines3d(rbind(AlignMeshesC$alset[2,],AlignMeshesC$alset[1,]),lwd=3,col="green")
  lines3d(rbind(AlignMeshesC$alset[2,],AlignMeshesC$alset[1,]),lwd=3,col="green")

  axes3d()
```

ToothCore *ToothCore*

Description

Tool to build 3D and 2D cross sections

Usage

```
ToothCore(
  out.sur = out.sur,
  inn.sur = inn.sur,
  num.sect = 61,
  bio.len,
  clean_int_out_O = TRUE,
  clean_int_out_I = TRUE,
  param1_out = 0.5,
  radius.fact_out = 2.5,
  param1_inn = 0.5,
  radius.fact_inn = 2.5,
  npovs = 100,
  num.points = 500,
  start = 0.2,
  end = 0.8,
  curved = FALSE,
```

```
  print.progress = TRUE
)
```

Arguments

out.sur	object of class mesh3d
inn.sur	object of class mesh3d
num.sect	number of sections
bio.len	length of the selected region of interest along with extracting the digital section
clean_int_out_O	logical if TRUE the outer section will be cleaned by using spherical flipping
clean_int_out_I	logical if TRUE the inner section will be cleaned by using spherical flipping
param1_out	numeric parameter for clean_int_out_O spherical flipping operator (how much the section will be deformed)
radius.fact_out	logical if TRUE the inner section will be cleaned by using spherical flipping
param1_inn	numeric parameter for clean_int_out_I spherical flipping operator (how much the section will be deformed)
radius.fact_inn	logical if TRUE the inner section will be cleaned by using spherical flipping
npovs	numeric: number of points of view defined around the section
num.points	number of equiengular 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
curved	logical: if TRUE the cutting planes will follow the mesh curvature (beta version)
print.progress	logical: if TRUE a progress bar is printed to the screen

Value

out3D num.pointsx3xnum.sect array of the external outlines

inn3D num.pointsx3xnum.sect array of the internal outlines

out3D num.pointsx2xnum.sect array of the external outlines

inn3D 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; Mathilde Augoyard

Examples

```
data("URI1_tooth")
require(morphomap)
require(Morpho)
require(rgl)
Enamel<-URI1_tooth$mesh1
Dentin<-URI1_tooth$mesh2
Pulp<-URI1_tooth$mesh3
outline<-URI1_tooth$outline
set<-URI1_tooth$set

#Unrolling the crown
AlignMeshes<-ToothAlignment(mesh1=Enamel,mesh2=Dentin,mesh3=Pulp,set,outline,analyse = "c")
#Virtual sectioning dentine-pulp
External<-AlignMeshes$almesh1$mesh
Internal<-AlignMeshes$almesh2$mesh
#Define 16 cross-sections from the 30% to the 90% along the crown
Core<-ToothCore(External,Internal,num.points = 1000,num.sect =16,
                bio.len = AlignMeshes$length,start=0.3,end=0.9)

plot3d(morphomapArray2matrix(Core$"out3D"),aspect=FALSE,xlab="x",ylab="y",zlab="z")
plot3d(morphomapArray2matrix(Core$"inn3D"),aspect=FALSE,add=TRUE)
wire3d(AlignMeshes$almesh2$mesh,col="white")
wire3d(AlignMeshes$almesh1$mesh,col="violet")

#Unrolling the rooth
AlignMeshes<-ToothAlignment(mesh1=Enamel,mesh2=Dentin,mesh3=Pulp,set,outline,analyse = "r")
#Virtual sectioning dentine-pulp
External<-AlignMeshes$almesh2$mesh
Internal<-AlignMeshes$almesh3$mesh
#Define 16 cross-sections from the 10% to the 50% along the root
Core<-ToothCore(External,Internal,num.points = 1000,num.sect =16,
                bio.len = AlignMeshes$length,start=0.1,end=0.5)

plot3d(morphomapArray2matrix(Core$"out3D"),aspect=FALSE,xlab="x",ylab="y",zlab="z")
plot3d(morphomapArray2matrix(Core$"inn3D"),aspect=FALSE,add=TRUE)
wire3d(AlignMeshes$almesh3$mesh,col="red")
wire3d(AlignMeshes$almesh2$mesh,col="lightblue")
wire3d(AlignMeshes$almesh1$mesh,col="white")
```

ToothDF *ToothDF*

Description

Tool to build a data.frame suitable for morphometric maps

Usage

```
ToothDF(
  tooth.thickness,
  rem.out = TRUE,
  fac.out = 0.5,
  smooth = TRUE,
  scale = TRUE,
  smooth.iter = 5,
  method = "equiangular",
  labels = c("Li", "Mes", "Bu", "D", "Li"),
  relThick = FALSE
)
```

Arguments

tooth.thickness	list: tooth.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 dental 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 dental thickness is calculated at each point as the closest distance between external and internal outlines
labels	character vector: names for x labels in the morphometric map
relThick	logical: if TRUE the dental thickness is converted into relative dental thickness

Value

XYZ data.frame for morphometric map

labels character vector for x labels in the morphometric map

Author(s)

Antonio Profico; Mathilde Augoyard

ToothExport *ToothExport*

Description

Export the output from ToothAlignement

Usage

```
ToothExport(input, id, file)
```

Arguments

input	list: output from ToothAlignement
id	character: label name
file	character: name the output file

Value

no return value, called for side effects (see description)

Author(s)

Antonio Profico; Mathilde Augoyard

ToothImport *ToothImport*

Description

Import the output from ToothAlignement

Usage

```
ToothImport(file)
```

Arguments

file	character: name of input file

Value

Mesh1: mesh of the aligned mesh1

Mesh2: mesh of the aligned mesh2

Mesh3: mesh of the aligned mesh3

B.Length: length of the region of interest

Landmarks: landmark coordinates

Outline: outline coordinates

Margins: position of the margins along the outline

diamBL: bucco-lingual diameter

diamMD: mesio-distal diameter

Author(s)

Antonio Profico; Mathilde Augoyard

`ToothPCA` *ToothPCA*

Description

Perform the Principal Component Analysis on a list of tooth.shape objects

Usage

```
ToothPCA(
  mpShapeList,
  gamMap = FALSE,
  nrow = 120,
  ncol = 80,
  gdl = 250,
  rem.out = TRUE,
  scaleThick = FALSE,
  relThick = FALSE,
  fac.out = 1.5,
  method = "equiangular",
  scalePCA = TRUE
)
```

Arguments

`mpShapeList`	list: tooth.shape objects
`gamMap`	logical: if TRUE gamMap spline method is applied
`nrow`	numeric: number of rows if gamMap is TRUE

ncol	numeric: number of columns if gamMap is TRUE
gdl	numeric: degree of freedom (if gamMap is TRUE)
rem.out	logical: if TRUE outliers are removed
scaleThick	logical: if TRUE thickness values are scaled from 0 to 1
relThick	logical: if TRUE the thickness values are scaled by the diameter from the centroid to the external outline
fac.out	numeric: threshold to define an outlier observation
method	character: "equiangular" or "closest" to define the thickness from evenly spaced or closest semilandmarks between the external and internal outline
scalePCA	logical: indicate whether the variables should be scaled to have unit variance

Value

PCscores matrix of PC scores

PCs principal components

variance table of the explained Variance by the PCs

meanMap mean map

CorMaps maps of thickness used as input in the PCA

Author(s)

Antonio Profico; Mathilde Augoyard

Examples

```
### Example on the canine crown
data("UCcrown")
require(morphomap)
shapeList<-UCcrown
PCA<-ToothPCA(shapeList,gamMap = FALSE,scaleThick = TRUE,scalePCA = TRUE ,relThick = FALSE)

#gamMap set on TRUE
PCA<-ToothPCA(shapeList,gamMap = TRUE,scaleThick = TRUE,scalePCA = TRUE,relThick = FALSE)

otu<-substr(names(shapeList),1,2)
pchs <- ifelse(otu == "MH", 16, 17)
cols <- ifelse(otu == "MH", "orange", "darkblue")

plot(PCA$PCscores,col=cols,cex=1, pch = pchs,
     xlab=paste("PC1 (",round(PCA$Variance[1,2],2),"%)"),
     ylab=paste("PC2 (",round(PCA$Variance[2,2],2),"%)"),
     cex.lab=1,cex.axis=1)
title (main="UC (radicular dentine)", font.main= 1,adj = 0, cex.main = 1.2)
legend("topright", legend = c("MH", "NE"), col = c("orange", "darkblue"), pch = c(16,17), cex = 0.8)
abline(v=0,h=0,col="black",lwd=2,lty=3)

hpts1 <- chull(PCA$PCscores[which(otu=="MH"),1:2])
hpts1 <- c(hpts1, hpts1[1])
```

```
polygon(PCA$PCscores[which(otu=="MH")[hpts1],1:2 ], col = adjustcolor("orange", 0.3), border = NA)

hpts2 <- chull(PCA$PCscores[which(otu=="NE"),c(1:2)])
hpts2 <- c(hpts2, hpts2[1])
polygon(PCA$PCscores[which(otu=="NE")[hpts2], ], col = adjustcolor("darkblue", 0.3), border = NA)

PC1min<-ToothVariations(PCA,min(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)
PC1max<-ToothVariations(PCA,max(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)

### Example on the canine root
data("UCroot")
require(morphomap)
shapeList<-UCroot
PCA<-ToothPCA(shapeList,gamMap = FALSE,scaleThick = TRUE,scalePCA = TRUE,relThick = FALSE)
otu<-substr(names(shapeList),1,2)
pchs <- ifelse(otu == "MH", 16, 17)
cols <- ifelse(otu == "MH", "orange", "darkblue")

plot(PCA$PCscores,col=cols,cex=1, pch = pchs,
     xlab=paste("PC1 (",round(PCA$Variance[1,2],2),"%)"),
     ylab=paste("PC2 (",round(PCA$Variance[2,2],2),"%)"),
     cex.lab=1,cex.axis=1)
title (main="UC (radicular dentine)", font.main= 1,adj = 0, cex.main = 1.2)
legend("topright", legend = c("MH", "NE"), col = c("orange", "darkblue"), pch = c(16,17), cex = 0.8)
abline(v=0,h=0,col="black",lwd=2,lty=3)

hpts1 <- chull(PCA$PCscores[which(otu=="MH"),1:2])
hpts1 <- c(hpts1, hpts1[1])
polygon(PCA$PCscores[which(otu=="MH")[hpts1],1:2 ], col = adjustcolor("orange", 0.3), border = NA)

hpts2 <- chull(PCA$PCscores[which(otu=="NE"),c(1:2)])
hpts2 <- c(hpts2, hpts2[1])
polygon(PCA$PCscores[which(otu=="NE")[hpts2], ], col = adjustcolor("darkblue", 0.3), border = NA)

PC1min<-ToothVariations(PCA,min(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)
PC1max<-ToothVariations(PCA,max(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)

### Example on the central upper incisor (crown)
data("UI1crown")
require(morphomap)
shapeList<-UI1crown
PCA<-ToothPCA(shapeList,gamMap = FALSE,scaleThick = TRUE,scalePCA = TRUE ,relThick = FALSE)
otu<-substr(names(shapeList),1,2)
pchs <- ifelse(otu == "MH", 16, 17)
cols <- ifelse(otu == "MH", "orange", "darkblue")

plot(PCA$PCscores,col=cols,cex=1, pch = pchs,
     xlab=paste("PC1 (",round(PCA$Variance[1,2],2),"%)"),
     ylab=paste("PC2 (",round(PCA$Variance[2,2],2),"%)"),
     cex.lab=1,cex.axis=1)
title (main="UC (radicular dentine)", font.main= 1,adj = 0, cex.main = 1.2)
legend("topright", legend = c("MH", "NE"), col = c("orange", "darkblue"), pch = c(16,17), cex = 0.8)
abline(v=0,h=0,col="black",lwd=2,lty=3)
```

```
hpts1 <- chull(PCA$PCscores[which(otu=="MH"),1:2])
hpts1 <- c(hpts1, hpts1[1])
polygon(PCA$PCscores[which(otu=="MH")[hpts1],1:2 ], col = adjustcolor("orange", 0.3), border = NA)

hpts2 <- chull(PCA$PCscores[which(otu=="NE"),c(1:2)])
hpts2 <- c(hpts2, hpts2[1])
polygon(PCA$PCscores[which(otu=="NE")[hpts2], ], col = adjustcolor("darkblue", 0.3), border = NA)

PC1min<-ToothVariations(PCA,min(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)
PC1max<-ToothVariations(PCA,max(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)

### Example on the upper central incisor (root)
data("UI1root")
require(morphomap)
shapeList<-UI1root
PCA<-ToothPCA(shapeList,gamMap = FALSE,scaleThick = TRUE,scalePCA = TRUE,relThick = FALSE)
otu<-substr(names(UI1root),1,2)
pchs <- ifelse(otu == "MH", 16, 17)
cols <- ifelse(otu == "MH", "orange", "darkblue")

plot(PCA$PCscores,col=cols,cex=1, pch = pchs,
     xlab=paste("PC1 (",round(PCA$Variance[1,2],2),"%)"),
     ylab=paste("PC2 (",round(PCA$Variance[2,2],2),"%)"),
     cex.lab=1,cex.axis=1)
title (main="UC (radicular dentine)", font.main= 1,adj = 0, cex.main = 1.2)
legend("topright", legend = c("MH", "NE"), col = c("orange", "darkblue"), pch = c(16,17), cex = 0.8)
abline(v=0,h=0,col="black",lwd=2,lty=3)

hpts1 <- chull(PCA$PCscores[which(otu=="MH"),1:2])
hpts1 <- c(hpts1, hpts1[1])
polygon(PCA$PCscores[which(otu=="MH")[hpts1],1:2 ], col = adjustcolor("orange", 0.3), border = NA)

hpts2 <- chull(PCA$PCscores[which(otu=="NE"),c(1:2)])
hpts2 <- c(hpts2, hpts2[1])
polygon(PCA$PCscores[which(otu=="NE")[hpts2], ], col = adjustcolor("darkblue", 0.3), border = NA)

PC1min<-ToothVariations(PCA,min(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)
PC1max<-ToothVariations(PCA,max(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)

### Example on the lateral upper incisor (crown)
data("UI2crown")
require(morphomap)
shapeList<-UI2crown
PCA<-ToothPCA(shapeList,gamMap = FALSE,scaleThick = TRUE,scalePCA = TRUE ,relThick = FALSE)
otu<-substr(names(shapeList),1,2)
pchs <- ifelse(otu == "MH", 16, 17)
cols <- ifelse(otu == "MH", "orange", "darkblue")

plot(PCA$PCscores,col=cols,cex=1, pch = pchs,
     xlab=paste("PC1 (",round(PCA$Variance[1,2],2),"%)"),
     ylab=paste("PC2 (",round(PCA$Variance[2,2],2),"%)"),
     cex.lab=1,cex.axis=1)
```

```
title (main="UC (radicular dentine)", font.main= 1,adj = 0, cex.main = 1.2)
legend("topright", legend = c("MH", "NE"), col = c("orange", "darkblue"), pch = c(16,17), cex = 0.8)
abline(v=0,h=0,col="black",lwd=2,lty=3)

hpts1 <- chull(PCA$PCscores[which(otu=="MH"),1:2])
hpts1 <- c(hpts1, hpts1[1])
polygon(PCA$PCscores[which(otu=="MH")[hpts1],1:2 ], col = adjustcolor("orange", 0.3), border = NA)

hpts2 <- chull(PCA$PCscores[which(otu=="NE"),c(1:2)])
hpts2 <- c(hpts2, hpts2[1])
polygon(PCA$PCscores[which(otu=="NE")[hpts2], ], col = adjustcolor("darkblue", 0.3), border = NA)

PC1min<-ToothVariations(PCA,min(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)
PC1max<-ToothVariations(PCA,max(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)

### Example on the upper lateral incisor (root)
data("UI2root")
require(morphomap)
shapeList<-UI2root
PCA<-ToothPCA(shapeList,gamMap = FALSE,scaleThick = TRUE,scalePCA = TRUE,relThick = FALSE)
otu<-substr(names(UI2root),1,2)
pchs <- ifelse(otu == "MH", 16, 17)
cols <- ifelse(otu == "MH", "orange", "darkblue")

plot(PCA$PCscores,col=cols,cex=1, pch = pchs,
     xlab=paste("PC1 (",round(PCA$Variance[1,2],2),"%)"),
     ylab=paste("PC2 (",round(PCA$Variance[2,2],2),"%)"),
     cex.lab=1,cex.axis=1)
title (main="UC (radicular dentine)", font.main= 1,adj = 0, cex.main = 1.2)
legend("topright", legend = c("MH", "NE"), col = c("orange", "darkblue"), pch = c(16,17), cex = 0.8)
abline(v=0,h=0,col="black",lwd=2,lty=3)

hpts1 <- chull(PCA$PCscores[which(otu=="MH"),1:2])
hpts1 <- c(hpts1, hpts1[1])
polygon(PCA$PCscores[which(otu=="MH")[hpts1],1:2 ], col = adjustcolor("orange", 0.3), border = NA)

hpts2 <- chull(PCA$PCscores[which(otu=="NE"),c(1:2)])
hpts2 <- c(hpts2, hpts2[1])
polygon(PCA$PCscores[which(otu=="NE")[hpts2], ], col = adjustcolor("darkblue", 0.3), border = NA)

PC1min<-ToothVariations(PCA,min(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)
PC1max<-ToothVariations(PCA,max(PCA$PCscores[,1]),PCA$PCs[,1],asp=0.5,meanmap = FALSE)
```

ToothRegradius *ToothRegradius*

Description

Finds equiangular landmarks

Usage

```
ToothRegradius(mat, center, n, direction = c("c", "a"))
```

Arguments

mat	a kx2 matrix
center	coordinates of the center from which the calculation of regular radius started
n	number of points
direction	specify direction: "c"=clockwise; "a"=anticlockwise

Value

V2 position of landmarks equiangular spaced

Author(s)

Antonio Profico; Mathilde Augoyard

ToothShape *ToothShape*

Description

Tool for the extraction of equiangular landmarks on the entire dental region of interest

Usage

```
ToothShape(
  tooth.core,
  num.land,
  sects_vector,
  cent.out = "E",
  delta = 0.1,
  direction = "c",
  out.sur = NULL,
  inn.sur = NULL
)
```

Arguments

tooth.core	list: tooth.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
direction	clockwise if "c", anticlockwise if"a"
out.sur	mesh: if provided, the external outlines will be projected back on the surface
inn.sur	mesh: if provided, the internal outlines will be projected back on the surface

Value

out3D num.pointsx3xnum.sect array in which the external outlines are stored

inn3D num.pointsx3xnum.sect array in which the internal outlines are stored

out2D num.pointsx2xnum.sect array in which the external outlines are stored

inn2D num.pointsx2xnum.sect array in which the interal 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 length of the selected region of interest

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

centroids of the cross-sections

Author(s)

Antonio Profico; Mathilde Augoyard

Examples

```
data("URI1_tooth")
require(morphomap)
require(rgl)
Enamel<-URI1_tooth$mesh1
Dentin<-URI1_tooth$mesh2
Pulp<-URI1_tooth$mesh3
outline<-URI1_tooth$outline
set<-URI1_tooth$set

#Unrolling the crown
AlignMeshes<-ToothAlignment(mesh1=Enamel,mesh2=Dentin,mesh3=Pulp,set,outline,analyse = "c")
#Virtual sectioning dentine-pulp
External<-AlignMeshes$almesh1$mesh
Internal<-AlignMeshes$almesh2$mesh
#Define 16 cross-sections from the 30% to the 90% along the crown
Core<-ToothCore(External,Internal,num.points = 1000,num.sect =16,
                bio.len = AlignMeshes$length,start=0.3,end=0.9)
Shape<-ToothShape(Core,num.land = 21,secs_vector = NULL,direction = "a")

plot3d(morphomapArray2matrix(Shape$"out3D"),type="s",radius = 0.1,aspect=FALSE,
xlab="x",ylab="y",zlab="z")
plot3d(morphomapArray2matrix(Shape$"inn3D"),type="s",radius = 0.1,aspect=FALSE,
add=TRUE)
```

```
wire3d(AlignMeshes$almesh2$mesh,col="white")
wire3d(AlignMeshes$almesh1$mesh,col="violet")

#Unrolling the rooth
AlignMeshes<-ToothAlignment(mesh1=Enamel,mesh2=Dentin,mesh3=Pulp,set,outline,
analyse = "r")
#Virtual sectioning dentine-pulp
External<-AlignMeshes$almesh2$mesh
Internal<-AlignMeshes$almesh3$mesh
#Define 16 cross-sections from the 10% to the 50% along the root
Core<-ToothCore(External,Internal,num.points = 1000,num.sect =16,
                bio.len = AlignMeshes$length,start=0.1,end=0.5)
Shape<-ToothShape(Core,num.land = 21,sects_vector = NULL,direction = "a")

plot3d(morphomapArray2matrix(Shape$"out3D"),type="s",radius = 0.1,aspect=FALSE,
xlab="x",ylab="y",zlab="z")
plot3d(morphomapArray2matrix(Shape$"inn3D"),type="s",radius = 0.1,aspect=FALSE,
add=TRUE)
wire3d(AlignMeshes$almesh3$mesh,col="red")
wire3d(AlignMeshes$almesh2$mesh,col="lightblue")
wire3d(AlignMeshes$almesh1$mesh,col="white")
```

`ToothThickness` *ToothThickness*

Description

Tool for the extraction of equiangular landmarks on the selected dental mesh

Usage

```
ToothThickness(tooth.shape)
```

Arguments

`tooth.shape`	list: tooth.shape object

Value

sect_thickness_eq dental thickness (equiangular method)

sect_thickness_cp dental thickness (closest point method)

sect_Rthickness_eq relative dental thickness (equiangular method)

sect_Rthickness_cp relative dental thickness (closest point method)

ALPM_thickness dental thickness at ALPM quadrants

tooth.shape tooth.shape object

Author(s)

Antonio Profico; Mathilde Augoyard

`ToothVariations` *ToothVariations*

Description

Calculate and return morphometric map variation

Usage

```
ToothVariations(
  PCA,
  scores,
  PC,
  asp = 2,
  pal = blue2green2red(101),
  meanmap = TRUE
)
```

Arguments

`PCA`	list: output from ToothPCA
`scores`	numeric: principal component value
`PC`	numeric: principal component eigenvalue
`asp`	numeric: x,y ratio of the morphometric map
`pal`	character vector: vector of colors
`meanmap`	logical: if TRUE the mean map corresponds to the real mean morphometric maps, otherwise the mean map is 0

Value

XYZ data.frame of morphometric map variation

Author(s)

Antonio Profico; Mathilde Augoyard

ToothVolumes *ToothVolumes*

Description

Extract volumes from the object ToothShape

Usage

```
ToothVolumes(
  ShapeExt,
  ShapeInn,
  col1 = "gray",
  col2 = "red",
  col3 = "green",
  alpha1 = 1,
  alpha2 = 1,
  alpha3 = 1,
  plot = FALSE
)
```

Arguments

ShapeExt	3D mesh: external mesh
ShapeInn	3D mesh: internal mesh
col1	color of the ShapeExt
col2	color of the ShapeInn
col3	color of the boolean operation between ShapeExt and ShapeInn
alpha1	value to set trasparancy of col1
alpha2	value to set trasparancy of col1
alpha3	value to set trasparancy of col1
plot	logical: if TRUE the volumes are shown

Value

meshOut: external selected mesh

meshInnT: internal selected mesh

meshDiff: differences between selected meshes

volumeT: volume of the external mesh

volInn:volume of the internal mesh

volDiff: difference between volumeT and volInn

Author(s)

Antonio Profico; Mathilde Augoyard

Examples

```
data("URI1_tooth")
require(morphomap)
Enamel<-URI1_tooth$mesh1
Dentin<-URI1_tooth$mesh2
Pulp<-URI1_tooth$mesh3
outline<-URI1_tooth$outline
set<-URI1_tooth$set

#Unrolling the crown
AlignMeshes<-ToothAlignment(mesh1=Enamel,mesh2=Dentin,mesh3=Pulp,set,outline,analyse = "c")
#Virtual sectioning dentine-pulp
External<-AlignMeshes$almesh1$mesh
Internal<-AlignMeshes$almesh2$mesh
#Define 16 cross-sections from the 30% to the 90% along the crown
Core<-ToothCore(External,Internal,num.points = 1000,num.sect =16,
                bio.len = AlignMeshes$length,start=0.3,end=0.9)
Shape<-ToothShape(Core,num.land = 100,sects_vector = NULL,direction = "a")
volumes<-ToothVolumes(Shape$"out3D",Shape$"inn3D",plot=TRUE)
unlist(volumes[4:6])
```

UCcrown *example dataset*

Description

list of ToothShape (modern human and neanderthal upper canines)

Usage

```
data(UCcrown)
```

Author(s)

Antonio Profico, Mathilde Augoyard

UCroot *example dataset*

Description

list of ToothShape (modern human and neanderthal upper canines)

Usage

```
data(UCroot)
```

Author(s)

Antonio Profico, Mathilde Augoyard

UI1crown *example dataset*

Description

list of ToothShape (modern human and neanderthal upper central incisors)

Usage

```
data(UI1crown)
```

Author(s)

Antonio Profico, Mathilde Augoyard

UI1root *example dataset*

Description

list of ToothShape (modern human and neanderthal upper central incisors)

Usage

```
data(UI1root)
```

Author(s)

Antonio Profico, Mathilde Augoyard

UI2crown *example dataset*

Description

list of ToothShape (modern human and neanderthal upper lateral incisors)

Usage

```
data(UI2crown)
```

Author(s)

Antonio Profico, Mathilde Augoyard

UI2root *example dataset*

Description

list of ToothShape (modern human and neanderthal upper lateral incisors)

Usage

```
data(UI2root)
```

Author(s)

Antonio Profico, Mathilde Augoyard

URI1_tooth *example dataset*

Description

3D dental meshes of a modern human upper right central incisor

Usage

```
data(URI1_tooth)
```

Author(s)

Antonio Profico, Mathilde Augoyard

Index