Package 'molaR'

April 7, 2016

Title Dental Surface Complexity Measurement Tools

Version 3.0

Description Surface topography calculations of Dirichlet's normal energy, relief index, and orientation patch count for teeth using scans of enamel caps. Importantly, for the relief index and orientation patch count calculations to work, the scanned tooth files must be oriented with the occlusal plane parallel to the x and y axes, and perpendicular to the z axis. The files should also be simplified, and smoothed in some other software prior to uploading into R.

Depends R (>= 2.10), alphahull, rgl, Rvcg

License ACM

LazyData true

Suggests knitr, rmarkdown, rglwidget

VignetteBuilder knitr

NeedsCompilation no

Author James D. Pampush [aut, cre, cph],
Julia M. Winchester [aut, cph],
Paul E. Morse [aut, cph],
Alexander Q. Vining [aut, cph]

Maintainer James D. Pampush <jdpampush@gmail.com>

RoxygenNote 5.0.1

Repository CRAN

Date/Publication 2016-01-13 22:53:00

R topics documented:

`bgplot3d_XQuartz` *function for correcting new XQuartz issues*

Description

crucial for plotting in Mac OS Yosemite and El Capitan

Usage

```
bgplot3d_XQuartz(expression)
```

Arguments

`expression`	expression calls from DNE3d RFI3d and OPC3d bgplot3d_XQuartz()

clustered_patches *A clustering function*

Description

This function gathers linked faces into patches

Usage

```
clustered_patches(Directional_Bin_Face_Pairs)
```

Arguments

Directional_Bin_Face_Pairs	the bins of face directions clustered_patches()

compute_energy_per_face *Function will compute the DNE per face.*

Description

This will generate each Dirichlet's normal energy for each triangular face on the surface.

Usage

```
compute_energy_per_face(plyFile)
```

Arguments

plyFile	a stanford PLY file compute_energy_per_face()

cSize *Centroid Size Function*

Description

Get the centroid size

Usage

```
cSize(x)
```

Arguments

x	point cloud cSize

Directional_Bins *This bins the faces into directional categories*

Description

bins into 8 directional categories on the basis of their orientations

Usage

```
Directional_Bins(plyFile, rotation = 0)
```

Arguments

plyFile	a stanford PLY file
rotation	the amount to rotate the specimen by Directional_Bins()

DNE *Calculate Dirichlet normal energy of a surface*

Description

A function that calculates Dirichlet normal energy following the method of Bunn et al. (2011) Comparing Dirichlet normal surface energy of tooth crowns, a new technique of molar shape quantification for dietary inference, with previous methods in isolation and in combination. Am J Phys Anthropol 145:247-261 doi: 10.1002 ajpa.21489

Usage

```
DNE(plyFile, outliers = 0.1)
```

Arguments

plyFile	An object of class 'mesh3d' and 'shape3d' with calculated normals
outliers	The percentile of Dirichlet energy density values to be excluded defaults to top 0.1 percent

Details

The function requires an object created by reading in a ply file utilizing either the read.ply or the read.AVIZO.ply function, with calculated normals.

Dirichlet normal energy is calculated on meshes that represent specimen surfaces and have already been simplified to 10,000 faces and pre-smoothed in a 3D data editing program.

The function does not include boundary vertices in the calculation, and therefore the analyzed surface cannot be closed (i.e., it must contain a hole). The function defaults to remove the top 0.1 percent of calculated energy densities as outliers. Mesh orientation does not affect for this calculation.

DNE3d *Plot results of a DNE analysis of a surface*

Description

plotting function

Usage

```
DNE3d(DNE_File, setRange = c(0, 0), logColors = TRUE, edgeMask = TRUE,
  outlierMask = TRUE, showEdgePts = FALSE, legend = TRUE,
  legendScale = 1, leftOffset = 0.75, fieldofview = 0)
```

Arguments

DNE_File	An object that stores the output of the DNE function
setRange	User-defined range for plotting color scheme, see Details
logColors	Logical that log transforms the color scheme
edgeMask	Logical that colors edge faces black to indicate their lack of contribution to the total Dirichlet normal energy
outlierMask	Logical that colors outlier faces dark gray to indicate their lack of contribution to the Dirichlet normal energy
showEdgePts	Logical that highlights the edge vertices in red to indicate their lack of contribution ot the total Dirichlet normal energy
legend	Logical indicating whether or not a legend shold be displayed
legendScale	numeric value setting the relative size of the legend similar in function to cex
leftOffset	numeric value between -1 and 1 setting the degree of offset for the plotted surface to the left. Larger values set further to right.
fieldofview	Passes an argument to par3d changing the field of view in degrees of the resulting rgl

Details

This function creates a heat map on the mesh surface corresponding to the Dirichlet normal energy of each face calculated by the DNE function. Hottest colors represent highest normal energy values

Dirichlet normal energies for the faces of a mesh surface tend to be positively skewed, with a small proportion of the faces contributing much of the total energy for the surface. When logColors is enabled the function colorizes based on the log transformed Dirichlet normal energies, allowing for finer resolution between faces near the mode of the energy per face distribution. Disabling logColors will display the untransformed Dirichlet normal energies.

The legend will update to reflect the other arguments chosen by the user. Colors currently display in the legend in bins, however the colors used in the displayed mesh surface are on a continuum. Ideally, the legend should reflect a continuous stretch of color from the lowest calculated Dirichlet normal energy to the highest. Future versions will adjust the legend to this more intuitive display.

By default, the function sets the lowest Dirichlet normal energy calculated among all faces to a cool color and the highest normal energy calculated among all faces to red, and then colors the remaining faces on a continuous color spectrum between these two end points using either absolute

or log transformed Dirichlet normal energy values (depending on the status of logColors). Since the scale is relative to the energies of the input surface, visual comparisons cannot directly be made between multiple plots of different surfaces. The setRange argument allows users to define the minimum and maximum of the plotting color scheme and use it in multiple plots. This enables the direct comparison of different surfaces to one another with red equal to the user-defined maximum and a cool color equal to the user-defined minimum. The user should choose reasonable bounds for the maximum and minimum that are near the maximum and minimum Dirichlet normal energies calculated for their surfaces. setRange will not accept negative values.

The leftOffset value sets how far to the left the surface will appear, intended to help avoid overlap with the legend. Defaults to 0.75.

legendScale sets the relative size of the scale in the same way cex works

fieldofview is set to a default of 0, which is an isometric projection. Increasing it alters the degree of parallax in the perspective view, up to a maximum of 179 degrees.

`DNE_Legend` *Make legend for DNE3d plot*

Description

plotting subfunction

Usage

```
DNE_Legend(start, end, colors, DNELabels, scaled = F, edgeMask = F,
  outlierMask = F, logColors = F, lineSize = 2, textSize = 1.75,
  rectSize = 1)
```

Arguments

`start`	value for the legend to start with, i.e. bottome value
`end`	value for the legend to end with, i.e. top value
`colors`	range of values, defaulting to heat colors
`DNELabels`	values for the labels
`scaled`	logical indicating whether the values are scaled
`edgeMask`	logical indicating whether of not edges are being masked and that information to be included in the legend
`outlierMask`	logical indicating whether outliers are masked
`logColors`	logical indicating colors are on log scale
`lineSize`	numerical value to determine line thickness in the legend
`textSize`	numerical value determinining the size of the text
`rectSize`	numerical value setting size of the legend box

Details

This is an internal function which builds a better DNE plot legend

The legend will reflect the elements used in the plot. This is an internal function. Users will have little need or call to interact with it. DNE_Legend()

edge_vertices *Function for finding the edge vertices*

Description

Function will sort through all the vertices of the surface and find the ones which are on the edge. This will be needed for identifying which should be masked and not inlcuded in the calculation of the final DNE value.

Usage

```
edge_vertices(plyFile)
```

Arguments

plyFile	a stanford PLY file
	edge_vertices()

Equal_Vertex_Normals *Important function for re-doing the vertex normals for the DNE calculation.*

Description

The geomorph import function does not generate the correct vertex normals.

Usage

```
Equal_Vertex_Normals(plyFile)
```

Arguments

plyFile	a stanford PLY file Equal_Vertex_Normals()

ex_tooth1 *4149_DU-LP-09_LM1*

Description

Lower M1 of a male mantled howler monkey, *Aloutta palliata*. Catalogue Number DU-LP 09

Usage

```
ex_tooth1
```

Format

A list of five objects, as follows:
vb, a 4 x 5118 dataframe.
it, a 3 x 10000 dataframe.
primitivetype, a character string
material, a NULL field
normals, a 4 x 5118 dataframe

Source

MorphoSource

References

http://www.morphosource.com/index.php/Detail/SpecimenDetail/Show/specimen_id/22

`ex_tooth2` *4147_DU-LP-07_LM1*

Description

Lower M1 of a female manteled howler monkey, *Aloutta palliata*. Catalogue Number DU-LP 07

Usage

```
ex_tooth1
```

Format

A list of five objects, as follows:
vb, a 4 x 5135 dataframe.
it, a 3 x 9997 dataframe.
primitivetype, a character string
material, a NULL field
normals, a 4 x 5135 dataframe

Source

MorphoSource

References

http://www.morphosource.com/index.php/Detail/SpecimenDetail/Show/specimen_id/29

face_areas *Function to calculate face areas.*

Description

This function calculates the area of each face on a ply file

Usage

```
face_areas(plyFile)
```

Arguments

plyFile	a stanford PLY file face_areas()

Face_Normals *Function to find Face Normals*

Description

This function re-computes the face normals in a way consistent with MorphoTester.

Usage

```
Face_Normals(plyFile)
```

Arguments

plyFile	a stanford PLY file Face_Normals()

index_paired_directed_faces *Index of paired faces with directions*

Description

This does some heavy lifting to pull together faces which are paired together. This is needed for many later functions for compiling OPC

Usage

```
index_paired_directed_faces(plyFile)
```

Arguments

plyFile	a stanford PLY file index_paired_directed_faces()

molaR_Batch *Run a batch of molaR analyses*

Description

A function which automats molaR analyses. User simply sets up the functions they want run and can leave the computer to do the rest.

Usage

```
molaR_Batch(pathname = getwd(), DNE = TRUE, RFI = TRUE, OPCr = TRUE,
  OPC = FALSE, Details = FALSE, DNE_outliers = 0.1, RFI_alpha = 0.01,
  OPCr_steps = 8, OPCr_stepSize = 5.626, OPCr_minimum_faces = 3,
  OPCr_minimum_area = 0, OPC_rotation = 0, OPC_minimum_faces = 3,
  OPC_minimum_area = 0)
```

Arguments

pathname	The path to the file containing all the PLY surfaces to be analyzed. Defaults to the working directory
DNE	logical indicating whether or not to perform DNE calculation Defaults to true
RFI	logical indicating whether or not to perform RFI calculation Defaults to true
OPCr	logical indicating whether or not to perform OPCr calculation Defaults to true
OPC	logical indicating whether or not to perform OPC calculation Defaults to false
Details	logical indicating whether or not to save the details of the RFI and OPCr calculations
DNE_outliers	the percentile at which outliers will be excluded is passed to the DNE function, defaults to 0.1
RFI_alpha	the size of the alpha passed to RFI function, defaults to 0.01
OPCr_steps	the number of steps the OPCr function should take, is passed to the OPCr function. Defaults to 8
OPCr_stepSize	the size of each rotation. Passed to the OPCr function. Defaults to 5.626 degrees
OPCr_minimum_faces	sets the lower boundary for number of faces a patch must have for inclusion in total count. Defaults to 3 or more.
OPCr_minimum_area	sets the lower boundary for percentage of the surface area a patch must make up for inclusion in the total patch count. Cannot be used with minimum_faces on. Defaults to zero
OPC_rotation	amount of rotation to apply during OPC calculation. Defaults to zero
OPC_minimum_faces	minimum number of faces a patch must contain to be counted in the OPC function. Defaults to 3.
OPC_minimum_area	minimum percentage of the surface area a patch must make up to be counted in the OPC function. Defaults to off

Details

This function allows a user to set the analyses from molaR they want to run, along with the specific parameters for each function and have a whole batch of PLY files analyzed and saved to a csv file. Function will perform analyses on all PLY files in the working directory or user can specify a file path.

`molaR_Clean` *Clean up problem ply files*

Description

Function will remove floating verticies, and faces with zero area. These can cause issues when using molaR's primary functions of DNE, RFI, and OPC

Usage

```
molaR_Clean(plyFile, cleanType = "Both")
```

Arguments

`plyFile`	An object of classes 'mesh3d' and 'shape3d'
`cleanType`	logical asking what to clean, Verticies, Faces or Both. Defaults to Both.

Details

This function cleans up problematic ply files. Some smoothed files will have faces of zero area, or floating verticies. DNE and OPC cannot be calculated on these files. Running the plys through this function will allow those calculations to be made.

`OPC` *Calculate orientation patch count of a surface*

Description

A function that bins patches of a mesh surface that share general orientation and sums the number of unique patches given certain parameters Modified into 3D from the original 2.5D method described by Evans et al. (2007) High-level similarity of dentitions in carnivorans and rodents.Nature 445:78-81 doi: 10.1038 nature05433

Usage

```
OPC(plyFile, rotation = 0, minimum_faces = 3, minimum_area = 0)
```

Arguments

`plyFile`	An object of classes "mesh3d" and "shape3d" with calculated normals
`rotation`	Rotates the file in degrees about the center vertical axis
`minimum_faces`	Minimum number of ply faces required for a patch to be counted towards the total patch count
`minimum_area`	Minimual percentage of total surface area a patch must occupy to be counted towards the total patch count

Details

The function requires a mesh object created by reading in a ply file utilizing either the read.ply, vcgPlyread, or read.AVIZO.ply function

Orientation patch count is calculated on meshes that represent specimen surfaces and have already been downsampled to 10,000 faces and pre-smoothed in a 3D data editing program. Alignment of the point cloud will have a large effect on patch orientation and must be done in a 3D data editing program such as Avizo, or using the R package auto3dgm prior to creating and reading in the ply file. The occlusal surface of the specimen must be made parallel to the X- and Y-axes and perpendicular to the Z-axis.

The default for minimum_faces is to ignore patches consisting of only a single face on the mesh. Changing the minimum_area value will disable minimum_faces.

`OPC3d` *Plot results of OPC analysis of a surface*

Description

A function that produces a three-dimensional rendering of face orientation on a surface. The OPC function will identify the orientations of mesh faces and assign them to patches. It must be performed prior to using the OPC3d function.

Usage

```
OPC3d(OPC_Output_Object, binColors = hsv(h = (seq(10, 290, 40)/360), s = 0.9,
  v = 0.85), patchOutline = FALSE, outlineColor = "black",
  maskDiscard = FALSE, legend = TRUE, legendScale = 1,
  legendTextCol = "black", legendLineCol = "black", leftOffset = 1,
  fieldofview = 0)
```

Arguments

`OPC_Output_Object`	An object that stores the output of the OPC function
`binColors`	Allows the user to change the colors filled in for each directional bin
`patchOutline`	logical whether or not to outline the patches
`outlineColor`	parameter designating which color to outline the patches in
`maskDiscard`	logical indicating whether to discard the unused patches
`legend`	Logical indicating whether or not a legend should be displayed
`legendScale`	cex style scaling factor for the legend
`legendTextCol`	parameter designating color for the legend text
`legendLineCol`	parameter designating the color for the legend lines
`leftOffset`	numeric parameters disginating how far to offset the surface
`fieldofview`	Passes an argument to par3d changing the field of view in dregrees of the resulting rgl window

Details

This function will assign a uniform color to all faces on the mesh surface that share one of the 8 orientations identified by the OPC function. The function returns a colored shade3d of the mesh so that patches can be visually inspected. Future versions will include the option to black out patches not included in the orientation patch count.

Several legend plotting options are availble including customizing the line and text colors using color names with legendTextCol and legendLineCol, both default to black. legendScale works like cex for setting the size of the relative size of the legend.

leftOffset will determine how far the plotted surface is moved to the left to avoid obstructing the legend. Users shold choose between -1 and 1.

fieldofview is set to a default of 0, which is an isometric projection. Increasing it alters the degree of parallax in the perspective view, up to a maximum of 179 degrees.

colors will support any vector of 8 colors, in any coloration scheme. Default draws from the hsv color space to evenly space color information, however user can supply a list of RGB values, character strings, or integers in place.

OPCr *Calculate average orientation patch count after several rotations*

Description

A function that calls OPC iteratively after rotating mesh a selected number of degrees around the Z-axis following Evans and Jernvall (2009) Patterns and constraints in carnivoran and rodent dental complexity and tooth size. J Vert Paleo 29:24A

Usage

```
OPCr(plyFile, Steps = 8, stepSize = 5.625, minimum_faces = 3,
  minimum_area = 0)
```

Arguments

plyFile	An object of classes 'mesh3d' and 'shape3d' with calculated normals
Steps	Number of iterations to run the OPC function on the mesh
stepSize	Amount of rotation in degrees about the Z-axis to adjust mesh surface by between each iteration of OPC
minimum_faces	Argument to pass to the OPC function
minimum_area	Argument to pass to the OPC function

Details

The function requires an object created by reading in a ply file utilizing either the read.ply or the read.AVIZO.ply function, with calculated normals.

Default number of Steps is 8, with a stepSize of 5.625 degrees, following the original definition of OPCR.

See the details for the OPC function for more information about preparing mesh surfaces and the effects of minimum_faces and minimum_area.

OPC_Legend *function for building a legend in OPC plots*

Description

crucial graphics subfunction

Usage

```
OPC_Legend(binColors = c(1:8), binNumber = 8, maskDiscard = F,
  lineSize = 2, textSize = 1.75, circSize = 1, textCol = "black",
  lineCol = "black")
```

Arguments

binColors	number sequence for bins and their colors
binNumber	numeric number of different bins
maskDiscard	logical determining whether faces will be blacked out because they are discarded
lineSize	numeric determining the thickness of the legend lines
textSize	numeric determining the size of the font
circSize	numeric for plotting size of the circle in pie chart legend
textCol	color for the text in the circle legend
lineCol	color for the lines in the legend OPC_Legend()

patches_for_each_direction
Function for gathering the patches for each direction

Description

This function will gather the patches in each of the 8 bins and ready it for patches_for_each_direction()

Usage

```
patches_for_each_direction(indexed_pairs)
```

Arguments

indexed_pairs	Pairs of touching faces

patches_per *A function for patches within each face*

Description

this gets some important information out of each patch

Usage

```
patches_per(patch_details, plyFile, minimum_faces = 3, minimum_area = 0)
```

Arguments

patch_details	information on each patch
plyFile	a stanford PLY file
minimum_faces	minimum number of faces in each counted patch
minimum_area	minimum area for a patch to be counted patches_per()

patch_details *Function for gathering patch details for each Orientation patch*

Description

This function does some simple math to lets us know about the patches

Usage

```
patch_details(clusterlist, plyFile)
```

Arguments

clusterlist	a list of faces in the cluster patch_details()
plyFile	a stanford PLY file

read.AVIZO.ply *Read mesh data from ply files saved by AVIZO*

Description

A function that reads Stanford ply files as saved by the 3D data visualization software Avizo

Usage

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

Arguments

file	An ASCII PLY file generated by Avizo
ShowSpecimen	Logical indicating whether or not the mesh should be displayed
addNormals	Logical indicating whether or not normals of mesh vertices shold be calculated and appended to object

Details

If ShowSpecimen is True, a gray shade3d of the mesh is generated in a new rgl window for previewing the specimen. When saving to the ply file type, Avizo inserts additional property parameters into the file heading that sometimes describe various components of the mesh. These additional properties cause the read.ply function native to the geomorph package to fail. This function properly reads ply files generated by Avizo (like read.ply) and can be stored as an object accepted as input in the other molaR functions. Ply files generated through other software (such as MeshLab) can be read using read.ply.

remove_boundary_faces *Remove boundary faces*

Description

Important function for masking the edge faces

Usage

```
remove_boundary_faces(Energy_Per_Face_Values, plyFile)
```

Arguments

Energy_Per_Face_Values	information on E per face remove_boundary_faces()
plyFile	a stanford PLY file

remove_outliers *Mask outliers on some faces*

Description

This function will block out the top 0.1 percent of the faces

Usage

```
remove_outliers(Energy_values, X)
```

Arguments

Energy_values	energy density values on faces
X	percentile above which to remove remove_outliers()

RFI *Calculate Boyer's (2008) relief index for a surface*

Description

A function that calculates relief index following Boyer (2008) Relief index of second mandibular molars is a correlate of diet among prosimian primates and other mammals. J Hum Evol 55:1118-1137 doi: 10.1016/j.jhevol.2008.08.002

Usage

```
RFI(plyFile, alpha = 0.01)
```

Arguments

plyFile	An object of classes 'mesh3d' and 'shape3d'
alpha	Step size for calculating the outline. See details

Details

The function requires an object created by reading in a ply file utilizing either the read.ply or the read.AVIZO.ply function, with calculated normals.

Relief index is calculated by the ratio of three-dimensional surface area to two dimensional area on meshes that represent specimen surfaces and have already been pre-smoothed in a 3D data editing program. Alignment of the point cloud will have a large effect on patch orientation and must be done in a 3D data editing program or auto3dgm prior to creating and reading in the ply file. The mesh must be oriented such that the occlusal plane is parallel to the X- and Y-axes and perpendicular to the Z-axis.

Some files may fail with pancake[TempF,] : subscript out of bounds. In these files it may be necessary to increase the alpha value which is default set to 0.01. Increasing the alpha value can cause the RFI function to over-estimate the size of the footprint. Caution should be exercised when troubleshooting by adjusting alpha

RFI3d *Plot 3D and 2D areas of a mesh used to calculate relief index*

Description

A function that plots a three-dimensional model of the mesh surface and includes a footprint of the two-dimensional area for visual comparison.

Usage

```
RFI3d(RFI_Output, displacement = -1.9, SurfaceColor = "gray",
  FootColor = "red", FootPts = FALSE, FootPtsColor = "black",
  Opacity = 1, legend = F, legendScale = 1, leftOffset = 0,
  fieldofview = 0)
```

Arguments

RFI_Output	An object that stores the output of the RFI function
displacement	Moves the surface footprint some proportion of the height of the mesh. 0 is no displacement. Expects a value, negative values displace the footprint downward.
SurfaceColor	changes the color of the 3D surface mesh
FootColor	changes color of the 2D surface footprint
FootPts	logical indicating whether to plot the flattened points of the footprint from the original ply file
FootPtsColor	color for the plotted footprint points
Opacity	adjusts the opacity of the 3D mesh surface
legend	Logical indicating whether or not to include a legend of the colors chosen to represent the 3D surface and footprint
legendScale	cex style numeric relative scaling factor for the legend
leftOffset	how numeric between -1 and 1 for which to offset the surface relative to the legend.
fieldofview	Passes an argument to par3d changing the field of view in degrees of the resulting rgl window

Details

This function can help to visualize the three-dimensional and two dimensional areas that are used in calculating the relief index of a surface by displaying both at the same time. The RFI function must be performed first.

Opacity can be adjusted in a range from fully opaque (1) to fully transparent (0) in order to help visualize the footprint. The vertical placement of the footprint along the Z axis can be altered with displace depending on how the user wishes to view the surface, or on the original mesh orientation.

fieldofview is set to a default of 0, which is an isometric projection. Increasing it alters the degree of parallax in the perspective view, up to a maximum of 179 degrees.

RFI_Legend *function for building a legend for RFI*

Description

crucial plotting subfunction for RFI3d

Usage

```
RFI_Legend(surfCol = "gray", footCol = "red", lineSize = 2,
  textSize = 1.75, legSize = 1, opac = 1)
```

Arguments

surfCol	color for the 3D surface defaults to gray
footCol	color for the 2D footprint defualts to red
lineSize	numeric for setting size of the line for legend
textSize	numeric for setting the size of the text in the legend works like cex
legSize	sets relative size of legend
opac	sets the value for the opacity of the tooth surface when that is engaged RFI_Legend()

tr *Trace function*

Description

Matrix algebra

Usage

```
tr(m)
```

Arguments

m	a square matrix tr()

vertex_to_face_list *function for making a list of faces on each vertex*

Description

crucial function for getting a list of faces which will gather the faces per vertex.

Usage

```
vertex_to_face_list(plyFile)
```

Arguments

plyFile	a stanford PLY file vertex_to_face_list()

Index