Package ‘GoodFibes’

August 18, 2023

Type Package

Title Detection and Reconstruction of Muscle Fibers from diceCT Image Data

Version 0.1.10

Date 2023-08-18

Description Reconstruction of muscle fibers from image stacks using textural analysis. Includes functions for tracking, smoothing, cleaning, plotting and exporting muscle fibers. Also calculates basic fiber properties (e.g., length and curvature).

License GPL (>= 2)

Depends R(>= 3.5.0), imager

Imports rgl, stats, graphics, grDevices, concaveman, prodlim, splines2

NeedsCompilation no

Author Jessica Arbour [aut, cre] (<https://orcid.org/0000-0003-4506-158X>)

Maintainer Jessica Arbour <jessica.arbour@mtsu.edu>

Repository CRAN

Date/Publication 2023-08-18 18:02:37 UTC

R topics documented:

GoodFibes-package 2
ant.final 3
ant.raw 4
check.overlap 5
color.scale 6
crop.stack 7
equalize.stack 8
fiber.angle 9
fiber.curve 11
fiber.lengths 12
fibers.smoothed 13
fuse.fibers 14
fuse.fibers.auto 15

good.fibes 16
muscle.plot 19
muscle.plot.multi 20
muscle.plot.stl 21
pointsGenerator 23
quality.check 23
sequencePlot 25
thresholdPlot 26

Index **28**

`GoodFibes-package` *Detection and Reconstruction of Muscle Fibers from diceCT Image Data*

Description

Reconstruction of muscle fibers from image stacks using textural analysis. Includes functions for tracking, smoothing, cleaning, plotting and exporting muscle fibers. Also calculates basic fiber properties (e.g., length and curvature).

Details

The DESCRIPTION file:

Package:	GoodFibes
Type:	Package
Title:	Detection and Reconstruction of Muscle Fibers from diceCT Image Data
Version:	0.1.10
Date:	2023-08-18
Authors@R:	person("Jessica", "Arbour", , "jessica.arbour@mtsu.edu", role = c("aut", "cre"), comment = c(ORCID = "0000
Description:	Reconstruction of muscle fibers from image stacks using textural analysis. Includes functions for tracking, smo
License:	GPL (>=2)
Depends:	R(>= 3.5.0), imager
Imports:	rgl, stats, graphics, grDevices, concaveman, prodlim, splines2
Author:	Jessica Arbour [aut, cre] (<https://orcid.org/0000-0003-4506-158X>)
Maintainer:	Jessica Arbour <jessica.arbour@mtsu.edu>

Index of help topics:

```
GoodFibes-package       Detection and Reconstruction of Muscle Fibers
                        from diceCT Image Data
ant.final               Ant muscle fibers finalized
ant.raw                 Raw ant muscle fibers
check.overlap           Check if muscle fibers are redundant
color.scale             A simple wrapper to create colors for a
                        continuous variable
```

`crop.stack`	Automated imaging cropping
`equalize.stack`	Automated histogram equalization of image state
`fiber.angle`	*Experimental!* Calculating the orientation of muscle fibers
`fiber.curve`	Calculate the relative curvature of muscle fibers
`fiber.lengths`	Calculated the length of smoothed muscle fibers
`fibers.smoothed`	Smoothing of estimated fiber paths
`fuse.fibers`	Fuse incomplete fibers
`fuse.fibers.auto`	Automated repeated fusing of fiber paths
`good.fibes`	Automated detection of muscle fibers from diceCT scans
`muscle.plot`	Plot a single muscle fiber
`muscle.plot.multi`	Plot multi muscle fibers with a muscle outline
`muscle.plot.stl`	Plot and export muscle fibers to stl
`pointsGenerator`	Line points generator
`quality.check`	Quality testing of possible muscle fibers detected by good.fibes
`sequencePlot`	Plot the path of a muscle fiber generated using good.fibes
`thresholdPlot`	Plot image from diceCT stack using thresholding

Functions in this package allow for the reconstruction of muscle fibers from diceCT image stacks. Fibers are detected using textural analysis, smoothed using splines and processed for path quality (grayscale variation). Muscle fibers can be plotted in 3D with or without the overall muscle outline, and the 3D muscle fibers can be exported as an STL file. Basic fiber architecture metrics can be calculated.

Author(s)

NA

Maintainer: NA

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

`ant.final` *Ant muscle fibers finalized*

Description

Muscle fibers reconstructed from the mandibular muscle of an ant *(Monomorium pharaonis)* (post processing).

Format

A list with 198 entries, each representing a muscle fiber reconstructed from an image stack. Each entry contains $fiber.points, the raw reconstructed fiber paths, and $fiber.smoothed, the smoothed paths.

Details

These fibers were reconstructed using *good.fibes*, checked for poor fiber paths using *quality.check*, and had fibers fused and merged using *fuse.fibers* and *check.overlap*. Also see "ant.raw" for the initial fiber paths from good.fibes.

Source

Fiber paths were generated from the ant dataset in Katzke et al (2022). Original image files available upon request.

References

Katzke, J., Puchenkov, P., Stark, H., and Economo, E. 2022. A Roadmap to Reconstructing Muscle Architecture from CT Data. *Integrative Organismal Biology* 4(1): 1-16.

Examples

```
data(ant.final)
fl<-fiber.lengths(ant.final, res = 0.000673107, df = 1)
```

`ant.raw` *Raw ant muscle fibers*

Description

Preliminary muscle fibers detected from the mandibular muscle of an ant *(Monomorium pharaonis)*. No quality checking or processing.

Usage

```
data("ant.raw")
```

Format

A list with 247 entries, each representing a muscle fiber reconstructed from an image stack (dataset available in examples below) and a partial, subsampled stack is available in extdata. Each entry contains $fiber.points, the raw reconstructed fiber paths, and $fiber.smoothed, the smoothed paths.

Details

These are the initial possible fibers detected using `good.fibes`. Also see *ant.final* for the processed fibers.

Source

Fiber paths were generated from the ant dataset in Katzke et al (2022). Original image files are stored under extdata (and see example).

References

Katzke, J., Puchenkov, P., Stark, H., and Economo, E. 2022. A Roadmap to Reconstructing Muscle Architecture from CT Data. *Integrative Organismal Biology* 4(1): 1-16.

Examples

```
data(ant.raw)
fl<-fiber.lengths(ant.raw, res = 0.000673107, df = 1)

#### this downloads the ant dataset image stack to a temp folder
olddir<-getwd()
setwd(tempdir())
download.file(url=
"https://github.com/jessica-arbour/Ant-Muscle-Image-Stack/raw/main/Ant_data.zip",
destfile="antdata.zip")

unzip("antdata.zip")
setwd(paste0(getwd(),"/Ant data"))

setwd(olddir)
```

`check.overlap` *Check if muscle fibers are redundant*

Description

Determines whether muscle fibers are likely to be repeats. The function compares pairs of fibers and determines 1) whether they are within min.vox of each other, and 2) if the average distance between them stays less than min.vox. If so the longer fiber is kept and the redundant fiber is dropped.

Usage

```
check.overlap(fiber.list, min.vox, df = 2)
```

Arguments

`fiber.list`	A list containing elements with $fiber.points. Generated by `good.fibes`, `fibers.smoothed`, `quality.check`, or `fuse.fibers`.
`min.vox`	The distance between fibers in voxels (pixels) for fibers to be considered redundant
`df`	The degree of curvature for spline interpolation via splines::ns

Value

`drop.fibers`	the index for the redundant fibers to be removed
`overlapping.fibers`	a matrix with the compared fibers, the index of which were kept and which were removed in each comparison
`fibers.removed`	a list with fiber.points with the redundant fibers excluded

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

`fuse.fibers`,`quality.check`,`good.fibes`

`color.scale` *A simple wrapper to create colors for a continuous variable*

Description

For any continuous variable like fiber length, produces a vector that can be used in plotting functions for the *col* argument. Uses colorRampPalette.

Usage

```
color.scale(fl, color1, color2)
```

Arguments

`fl`	A numeric vector containing a measurement for each fiber in a fiber list.
`color1`	A character value for the color for small values of the variable
`color2`	A character value for the color for large values of the variable

Value

A vector with color values for each fiber

See Also

`muscle.plot.stl`

Examples

```
data(ant.final)
fl<-fiber.lengths(ant.final, res = 0.000673107, df=1)

cols<-color.scale(fl, "blue", "green")
muscle.plot.stl(ant.final, res = 0.000673107, cols = cols, mirror.axis = TRUE, df=1)
```

`crop.stack` *Automated imaging cropping*

Description

Automatically crops a stack of png image files to the minimum bounds of non-black values. Or if bounds are supplied can be cropped to the exact size of another stack of images.

Usage

```
crop.stack(images, bounds = NULL, save.images=FALSE)
```

Arguments

`images`	A vector with file names for the image files, can be created with list.files.
`bounds`	An optional vector with four values, given as c(xlim, xmax, ylim, ymax). These are printed at the end of the function when cropping is performed automatically.
`save.images`	When TRUE, images are saved to the current directory using imager::save.image.

Value

The bounds to be used for cropping. Optionally crops and saves images to working folder.

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

equalize.stack

Examples

```
olddir<-getwd()

#### this downloads the ant dataset image stack
#### if you have it already downloaded you can navigate to that folder
setwd(tempdir())
download.file(url=
"https://github.com/jessica-arbour/Ant-Muscle-Image-Stack/raw/main/Ant_data.zip",
destfile="antdata.zip")

unzip("antdata.zip")
setwd(paste0(getwd(),"/Ant data"))
####

images<-list.files(pattern=".png")

crop.stack(images)

setwd(olddir)
```

`equalize.stack` *Automated histogram equalization of image state*

Description

Conducts histogram equalization to adjust the contrast of the image stack. May improve visibility of muscle fibers prior to fiber detection. Optionally automatically save new image stack in working directory.

Usage

```
equalize.stack(images, n, save.images = FALSE)
```

Arguments

`images`	A vector of png image file names, created using list.files
`n`	The number of the image in the stack to be equalized and plotted
`save.images`	Should the whole image stack be equalized and plotted?

Value

Creates a plot and optionally saves an image stack

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans

See Also

`crop.stack`

Examples

```

olddir<-getwd()

#### this downloads the ant dataset image stack
#### if you have it already downloaded you can navigate to that folder
setwd(tempdir())
download.file(url=
"https://github.com/jessica-arbour/Ant-Muscle-Image-Stack/raw/main/Ant_data.zip",
destfile="antdata.zip")

unzip("antdata.zip")
setwd(paste0(getwd(),"/Ant data"))
####

images<-list.files(pattern=".png")
equalize.stack(images, 100)


setwd(olddir)
```

`fiber.angle` **Experimental!** *Calculating the orientation of muscle fibers*

Description

Calculating the angle of individual muscle fibers. Currently this function can only calculate fiber angles around a central axis (x, y or z). Fiber data is centered around the other axes.

Will eventually include an option for a tendon image stack. Not yet complete

Usage

```
fiber.angle(fib.list, axis, centered = TRUE)
```

Arguments

`fib.list`	A list of muscle fiber paths generated by good.fibes or from the various cleaning and processing function (must contain $fiber.points).
`axis`	The axis around which angles will be calculated as a deviation from
`centered`	Should the data be centered on the other two axes before calculation.

Value

A vector of angles in degrees corresponding to each fiber in the original list

Note

NOT TOTALLY VERIFIED YET, USE WITH CAUTION

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

Katzke, J., Puchenkov, P., Stark, H., and Economo, E. 2022. A Roadmap to Reconstructing Muscle Architecture from CT Data. *Integrative Organismal Biology* 4(1): 1-16.

Sullivan, S., McGechie, F., Middleton, K., and Holliday, C. 3D Muscle Architecture of the Pectoral Muscles of European Starling (Sturnus vulgaris).*Integrative Organismal Biology* 1(1):1-18.

See Also

`fiber.lengths`

Examples

```
olddir<-getwd()

data(ant.final)
fangle<-fiber.angle(ant.final,3)
fangle

cols<-color.scale(fangle, "blue", "red")
muscle.plot.stl(ant.final, cols=cols, df = 1)

setwd(olddir)
```

`fiber.curve` *Calculate the relative curvature of muscle fibers*

Description

Calculates a metric for fiber curvature. This is the ratio between the total length of the curved smoothed fiber, to the straight line distance between the end points of the fiber. A straight fiber will have a curvature value of ~ 1 (small differences may be due to the calculation of fiber length across a smoothed curve), and values > 1 represent more curvature.

Optionally identified which fibers show unusual curvature (are outliers), for possible removal.

Usage

```
fiber.curve(fib.list, df, check = TRUE, length.out=500)
```

Arguments

`fib.list`	A list of fibers containing $fiber.points. Produced by good.fibes or the various cleaning functions (quality check, fuse.fibers, check.overlap)
`df`	Corresponds to the df argument in splines2::nsp. Determines the shape of the smoothing spline (df = 1 represents straight muscle fibers)
`check`	Should unusually curved fibers be identified?
`length.out`	The number of straight line segments that the smoothed curve will be divided into for calculation of length

Value

`curvature`	The ratio of fiber length to end-to-end length
`problem.fibers`	Fibers with unusually high curvature. Given as the index of these fibers in the original list.

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

`fiber.lengths`,`fiber.angle`

Examples

```
data(ant.final)

fcr<-fiber.curve(ant.final,df=2,check=TRUE)
#fibers reconstructed with a curve here merely to demonstrate function
#ant fibers were fairly straight

sort(fcr$curvature)
#all fibers are close to 1 even with a "curved" reconstruction
```

`fiber.lengths` *Calculated the length of smoothed muscle fibers*

Description

Determines the length of reconstructed and smoothed muscle fibers. Fibers are smoothed using splines::ns and then oversampled (length.out). The sum of all straight line segments on the smoothed paths is taken as the overall fiber length

Usage

```
fiber.lengths(fib.list, res = NULL, df = 2, length.out = 500)
```

Arguments

`fib.list`	A list of fibers containing $fiber.points. Produced by good.fibes or the various cleaning functions (quality check, fuse.fibers, check.overlap)
`res`	The resolution of the isometric voxels (i.e., the distance between images). Should be given as a linear measure (e.g., mm, um)
`df`	The degrees of freedom passed to splines2::nsp. A df = 1 produces a straight fiber, while values >1 allow fibers to curve.
`length.out`	The number of straight line segments that the smoothed curve will be divided into for calculation of length

Value

A vector with fiber lengths

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

`fiber.angle`,`fiber.curve`

Examples

```
data(ant.final)


fl<-fiber.lengths(ant.final, res = 0.000673107, df=1)
mean(fl)
```

`fibers.smoothed` *Smoothing of estimated fiber paths*

Description

Applies splines to smooth the stepwise fiber paths produced by good.fibes, with the function *ns* from **splines**.

Usage

```
fibers.smoothed(fib.list, df)
```

Arguments

`fib.list`	A list of fibers with $fiber.points, produced by `good.fibes` or one of the cleaning and processing functions (e.g., fuse.fibers).
`df`	The degrees of freedom passed to splines2::nsp. Knots equal to df - 1 - intercept are set as breakpoints in the spline curve. A straight line path has a df of 1.

Value

`fiber.points`	The original fiber path from good.fibes
`fiber.smoothed`	The curved, smoothed fiber paths

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

`good.fibes`

`fuse.fibers` *Fuse incomplete fibers*

Description

This function compares fibers that pass between a minimum number of voxels and determines if merging them into a single fiber produces a well supported path

Usage

```
fuse.fibers(fiber.list, min.vox, min.improvement = 0.25, df = 2, length.out = 100)
```

Arguments

`fiber.list`	A list of fibers with $fiber.points produced by good.fibes or any of the processing and cleaning functions.
`min.vox`	The voxel distance below which fibers will be compared. Should be <= the voxel width of the muscle fascicles, though lower if interstital spaces are low.
`min.improvement`	The minimum increase (as a proportion) in fiber length for fibers to be worth merging.
`df`	The df to be used in smoothing fiber paths in the calculation of fiber length
`length.out`	The number of divisions to be used in the calculation of fiber lengths (line segments)

Details

This function compares pairs of fibers if they come within mix.vox of each other along their path. The fibers will be merged if 1) the mean 3D residual from the new spline through the combined fiber path is less than the mean residual from the two separate fiber paths, and 2) the fiber length of the combined fiber is at least min.improvement (proportionately) greater than the previous fiber lengths.

Value

`merged.fibers`	A list of fibers with $fiber.points, with fibers combined based on above thresholds
`fibers.to.merge`	a matrix of pairs of fibers that were merged

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans

See Also

`quality.check`, `check.overlap`

`fuse.fibers.auto` *Automated repeated fusing of fiber paths*

Description

This function repeatedly compares pairs of muscle fibers for fusing. Repeated applications of fuse.fibers until no further fibers can be merged

Usage

```
fuse.fibers.auto(fiber.list, min.vox, min.improvement = 0.25,
df = 2, length.out = 50, max.iter = 10, verbose = FALSE)
```

Arguments

`fiber.list`	A list of fibers with $fiber.points produced by good.fibes or any of the processing and cleaning functions.
`min.vox`	The voxel distance below which fibers will be compared. Should be <= the voxel width of the muscle fascicles, though lower if interstital spaces are low.
`min.improvement`	The minimum increase (as a proportion) in fiber length for fibers to be worth merging.
`df`	The df to be used in smoothing fiber paths in the calculation of fiber length
`length.out`	The number of divisions to be used in the calculation of fiber lengths (line segments)
`max.iter`	The maximum number of iterations of fuse.fibers that will be attempted.
`verbose`	Should the number of iterations through the function be displayed while running?

Value

A list of fibers with $fiber.points

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans

See Also

`fuse.fibers`

`good.fibes` *Automated detection of muscle fibers from diceCT scans*

Description

The function good.fibes uses textural analysis to determine the path of muscle fibers/fascicles through an image stack from a iodine contrast CT scan. Fiber paths are reconstructed using a stepwise algorithm that follows paths of low variation in threshold values. See details for full description of the method

SOMETIMES ABORTS RSTUDIO BUT STILL RUNS IN R-GUI, PROBLEM IS ONLY IN RSTUDIO

Usage

```
good.fibes(images, zero.image, radius, threshold = NULL,
cutoff, scaler = 1, blackcut = 0.95, seeds = 1, show.plot = TRUE,
start.seed = NULL, allowed.black = 0, bound.buffer = 0, backstep = 0, verbose=TRUE)
```

Arguments

`images`	A character vector with image names representing the image stack from a diceCT scan. The voxels are assumed to be isometric, and the images should be in .png format. The vector can be produced using list.files(pattern = ".png")
`zero.image`	The number of the image in the stack from which seed points should be drawn. Only one image can be selected.
`radius`	The number of images to consider forward or backward from the zero.image at each step of the walk. Maximum 11.
`threshold`	The grayscale value below which voxels will be considered black for the selection of seed points. Must be equal to or greater than cutoff
`cutoff`	The grayscale value below which voxels will be considered black in the forwards and backwards walk. Use `thresholdPlot` to determine a value that isolates muscle voxels from other tissues/background noise.
`scaler`	Exponential scaler for the trajectory penalization. Default is 1. At a value of 0 there is no trajectory penalization
`blackcut`	A termination condition. If a specified percentage (as proportion, e.g., 0.95) of voxels in the hemisphere of paths are black, the algorithm will terminate.
`seeds`	The number of seed points on the starting image. The seed points will generate a possible fiber path, if a walk is possible (seeds can fail if they are located on noise without possible paths).
`show.plot`	Optionally show the location of the tracker in the image stack at each step
`start.seed`	Optionally applies *set.seed* in the tracker to make results reproducible from one run to the next. See set.seed for more details.
`allowed.black`	For noisy datasets, allows this number of voxels with grayscale values below *cutoff* to be included in the possible paths without terminating the algorithm

`bound.buffer`	If a fiber path is within this many voxels distance of the outermost boundary of the muscle, as determined by grayscale values above the *cutoff*, then the algorithm will terminate the particular path. See details.
`backstep`	How many images "behind" the current plane should be considered. Should be kept to low values (1-3). When backstep = 0, only paths ending on images "ahead" of the image plane will be considered. **EXPERIMENTAL**, will create some weird paths. Use only if the muscle fibers definitely arc back through the image stack and perhaps only on image planes close to that point.
`verbose`	If TRUE will list the progress through each fiber

Details

The function begins by selecting a set of seed points from the selected image. Grayscale values below *threshold* are excluded, and the pairwise euclidean distances among all remaining voxels are calculated. Cluster analysis is conducted using *hclust* and a set of groups equal to *seeds* are produced using *cutree*. Voxels within each group are randomly selected.

From a selected seed voxel, a hemisphere of possible paths is projected, extending *radius* images from the selected starting image. If *backstep* is >0, paths within the starting image plane and behind the plane (1 = 1 image behind, 2 = 2 images behind) are also included in the possible paths. **NOTE** backstep is experimental and does cause more circuitous paths, use only if fiber paths reverse direction through the image stack at some point.

The forward walk from the seed point begins by choosing from the set of possible paths, the one that minimizes the following function.

diagnostic value = scaled grayscale SD * trajectory ^ scaler

1)The scaled grayscale SD is the standard deviation of grayscale values along each possible path. This value is scaled to 0 to 1. 2)The trajectory is the straight line distance between the end points on the hemisphere between the previous step and the next possible steps. This value is scaled to a range of 0 to 1, and added to 1 (resulting values range from 1 to 2). This is to penalize steps that make severe changes, as muscle fibers tend to not have very severe bends. 3)The impact of the trajectory penalization is scaled using *scaler*. If scaler = 0, there is no trajectory penalization.

The path with the minimum diagnostic value is selected, and the process repeated from the end point of that path. This stepwise algorithm continues the forward walk through the image stack until one of several stop conditions is reached:

1) The only available remaining paths would either terminate or cross a black voxel. This prevents the tracker from passing out of the muscle fascicle. To accommodate noisy datasets, the tracker may be permitted to cross a small number of black voxels (*allowed.black*). 2) The number of black voxels in the possible paths exceeds a specified number (e.g., 95 percent). This is meant to isolate regions of noise towards the end of a muscle fascicle, where adjoining connective tissue may obscure the end of a fiber. 3) The remaining paths would terminate within a specified distance of the external "boundary" of the muscle. This prevents fibers from continuing to track along the exterior surface of the muscle in noisy image stacks.

Once the path is terminated, the algorithm returns to the seed point and begins a walk in the opposite direction. It proceeds using the same terms as above. The forwards and backwards walks are returned. The process repeats for the next seed point.

Value

A list with a length equal to or less than seeds (failed paths will be dropped). Each element contains $fiber.points, the 3D coordinates providing the fiber path through the image stack, expressed in units of voxels.

Can be combined with separate runs from other images planes using c(). See example

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

`fibers.smoothed`

Examples

```
olddir<-getwd()

#### this downloads the ant dataset image stack
#### if you have it already downloaded you can navigate to that folder
setwd(tempdir())
download.file(url=
"https://github.com/jessica-arbour/Ant-Muscle-Image-Stack/raw/main/Ant_data.zip",
destfile="antdata.zip")

unzip("antdata.zip")
setwd(paste0(getwd(),"/Ant data"))
####


images<-list.files(pattern=".png")

fibes1<-good.fibes(images = images, zero.image = 200, radius = 9, threshold = 0.7,
cutoff = 0.65, seeds=5, start.seed = 1, show.plot=FALSE)

fibes2<-good.fibes(images = images, zero.image = 300, radius = 9, threshold = 0.7,
cutoff = 0.65, seeds=5, start.seed = 1, show.plot=FALSE)

fibes<-c(fibes1,fibes2)

muscle.plot.multi(fibes, images, df=1)


setwd(olddir)
```

`muscle.plot` *Plot a single muscle fiber*

Description

Used to compare the muscle fiber path to the smoothed muscle fiber. Plots a single set of $fiber.points from and the smoothed fibers.

Usage

```
muscle.plot(fiber.dat, images, df = 4, mirror.axis = FALSE, outline = 50, size = 2)
```

Arguments

`fiber.dat`	Any set of $fiber.points produced by good.fibes
`images`	A character vector with image names representing the image stack, can be produced using list.files.
`df`	The df to be used in smoothing fiber paths in the calculation of fiber length
`mirror.axis`	Depending on the way the image stack was exported, fibers may be reflected from their original original. mirror.axis = TRUE will reflect the fibers before plotting to correct this
`outline`	The number of wireframe "outlines" to draw the muscle boundaries
`size`	point size for $fiber.points in plot

Value

Returns a 3D plot

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

`muscle.plot.multi`, `muscle.plot.stl`

Examples

```
olddir<-getwd()

#### this downloads the ant dataset image stack
#### if you have it already downloaded you can navigate to that folder
setwd(tempdir())
download.file(url=
"https://github.com/jessica-arbour/Ant-Muscle-Image-Stack/raw/main/Ant_data.zip",
destfile="antdata.zip")

unzip("antdata.zip")
setwd(paste0(getwd(),"/Ant data"))
####

images<-list.files(pattern=".png")

data(ant.final)
muscle.plot(ant.final[[100]]$fiber.points,images,df=1, outline=30, mirror.axis=TRUE)
setwd(olddir)
```

`muscle.plot.multi` *Plot multi muscle fibers with a muscle outline*

Description

Uses functions from rgl to plot all fibers (smoothed with splines) in a fiber list. Also uses grayscale values from the image stack to determine the external boundaries of the muscle based on concave hulls. Boundaries are plotted as a series of single outlines sampled across the image.

Usage

```
muscle.plot.multi(fiber.list, images, df = 2, outline = 30,
cols = NULL, mirror.axis = FALSE)
```

Arguments

`fiber.list`	A list of fibers with $fiber points. Generated by good.fibes or processed cleaned by other functions
`images`	A character vector of image stack file names. Generated with list.files
`df`	The degrees of freedom to pass to splines2::nsp for smoothing fiber paths. df = 1 gives a straight path, while >1 gives increasingly curved paths
`outline`	The number of wireframe "outlines" to draw the muscle boundaries
`cols`	An optional vector of colors, the same order and length of fiber.list
`mirror.axis`	Depending on the way the image stack was exported, fibers may be reflected from their original original. mirror.axis = TRUE will reflect the fibers before plotting to correct this

Value

Returns a 3D plot

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

`muscle.plot`, `muscle.plot.stl`

Examples

```
olddir<-getwd()

#### this downloads the ant dataset image stack
#### if you have it already downloaded you can navigate to that folder
setwd(tempdir())
download.file(url=
"https://github.com/jessica-arbour/Ant-Muscle-Image-Stack/raw/main/Ant_data.zip",
destfile="antdata.zip")

unzip("antdata.zip")
setwd(paste0(getwd(),"/Ant data"))
####

images<-list.files(pattern=".png")
data(ant.final)
muscle.plot.multi(ant.final, images, df=1, mirror.axis = TRUE)


setwd(olddir)
```

`muscle.plot.stl` *Plot and export muscle fibers to stl*

Description

Plot a series of muscle fibers produced by good.fibes. Fibers are smoothed using splines before plotting. Optionally export an STL file in the correct size scale.

Usage

```
muscle.plot.stl(fiber.list, res = 1, df = 2, radius = 1, cols = NULL,
save.plot = FALSE, file.name = "muscle.fibers.stl", mirror.axis = FALSE)
```

Arguments

`fiber.list`	A list of fibers with $fiber points. Generated by good.fibes or processed cleaned by other functions
`res`	The isometric resolution of the original scan (i.e., the distance between images). Provided as a linear measure (um, mm, etc.)
`df`	The degrees of freedom to pass to splines2::nsp for smoothing fiber paths. df = 1 gives a straight path, while >1 gives increasingly curved paths
`radius`	The radius of the lines plotted for muscle fibers
`cols`	An optional vector of colors, the same order and length of fiber.list
`save.plot`	When TRUE, plot is saved as an .stl object in the current working directory. Provide file in file.name argument.
`file.name`	Character data giving the file.name and ending in .stl
`mirror.axis`	Depending on the way the image stack was exported, fibers may be reflected from their original original. mirror.axis = TRUE will reflect the fibers before plotting to correct this

Value

Returns a 3D plot

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

`muscle.plot.multi`,`muscle.plot`,`good.fibes`

Examples

```
data(ant.final)

muscle.plot.stl(ant.final, res = 0.000673107, df=1, radius = 1,
mirror.axis = TRUE, save.plot = FALSE)
```

`pointsGenerator` *Line points generator*

Description

An internal function for generating line coordinates for all possible paths of the good.fibes tracking algorithm.

Usage

```
pointsGenerator(startx, starty, ucoords, radius = radius, backstep)
```

Arguments

`startx`	seed point x coordinate
`starty`	seed point y coordinate
`ucoords`	unique end coordinates generated by hemisphere.points
`radius`	Number of images to consider forward from the zero.image
`backstep`	Should images in the seed point plane or behind it be considered. Each value (1, 2, 3, etc.) gives the number of images "behind" the seed point to consider in the spherical dome

Details

For internal use in good.fibes only

Value

A list containing coorindates for each line ending in the end points determined by hemisphere.points

---

`quality.check` *Quality testing of possible muscle fibers detected by good.fibes*

Description

Calculates quality as the ratio of grayscale standard deviation and fiber length for each muscle fiber detected using good.fibes. Long, homogenous fibers are considered to be of higher quality. Fibers are smoothed before the calculation of fiber quality.

Fibers with usually low quality (high grayscale variation compared to fiber length) are identified for exclusion.

Usage

```
quality.check(fib.list, images, res, min.length = NULL, length.out = 200, df = 2)
```

Arguments

`fib.list`	A list of muscle fibers with $fiber.points, generated by good.fibes
`images`	A character vector of image stack file names. Generated with list.files
`res`	The isometric resolution of the voxels (i.e., the distance between images). Given as a linear measure (um, mm, etc.)
`min.length`	Optionally exclude fibers below a certain fiber length (e.g., based on anatomical measurements). If resolution is given, then in those units, otherwise in number of voxels.
`length.out`	Number of line segments used in the calculation of fiber length
`df`	Degrees of freedom passed to splines::ns in the smoothing of muscle fibers before calculation. df = 1 produces straight fibers, while values > 1 produce increasingly curved fibers

Value

`quality`	grayscale sd/fiber length, low values are considered of higher quality
`grey.values`	A list providing the grayscale values for each smoothed fibers
`problem.fibers`	The location of fibers in the original list object that have atypically poor quality and should be excluded from further analyses

Note

Also produces a plot showing the distribution of quality values, and numbered bars for outliers.

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

Puffel, F. Pouget, A., Liu, X., Zuber, M., van de Kamp, T., Roces, F., and Labonte, D., 2021. *Journal of the Royal Society Interface* 18: 20210424

See Also

`fuse.fibers`,`check.overlap`,

Examples

```
olddir<-getwd()

#### this downloads the ant dataset image stack
#### if you have it already downloaded you can navigate to that folder
setwd(tempdir())
```

```
download.file(url=
"https://github.com/jessica-arbour/Ant-Muscle-Image-Stack/raw/main/Ant_data.zip",
destfile="antdata.zip")

unzip("antdata.zip")
setwd(paste0(getwd(),"/Ant data"))
####

data(ant.raw)
images<-list.files(pattern=".png")

qc<-quality.check(ant.raw[21:50],images, res=0.000673107, df=1)


setwd(olddir)
```

`sequencePlot` *Plot the path of a muscle fiber generated using good.fibes*

Description

Plots images in sequence showing the image stack and the location of the muscle fiber path at each step in the fiber tracking algorithm in good.fibes

Usage

```
sequencePlot(fib.track, images, threshold = 0.1, sleep.time = 0.5)
```

Arguments

`fib.track`	A set of $fiber.points from a fiber list, generated by good.fibes
`images`	A character vector of image stack file names. Generated with list.files
`threshold`	A cutoff values under which voxels are set to black
`sleep.time`	Time in seconds between images, sets speed for plotting sequence

Value

Returns a sequence of plots

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

thresholdPlot

Examples

```
olddir<-getwd()

#### this downloads the ant dataset image stack
#### if you have it already downloaded you can navigate to that folder
setwd(tempdir())
download.file(url=
"https://github.com/jessica-arbour/Ant-Muscle-Image-Stack/raw/main/Ant_data.zip",
destfile="antdata.zip")

unzip("antdata.zip")
setwd(paste0(getwd(),"/Ant data"))
####

images<-list.files(pattern=".png")

data(ant.raw)

sequencePlot(ant.raw[[2]]$fiber.points, images, 0 ,0.2)



setwd(olddir)
```

`thresholdPlot` *Plot image from diceCT stack using thresholding*

Description

Plot a selected image from the image stack with values below *threshold* set to black (grayscale = 0). Can be used to select *threshold* and *cutoff* values used in good.fibes.

Usage

```
thresholdPlot(images, n, threshold)
```

Arguments

`images`	A character vector of image stack file names. Generated with list.files
`n`	The number of the selected image in the vector "images"
`threshold`	The cutoff value for grayscale values. All voxels with grayscales below threshold will be displayed as black.

Value

Returns a plot

Author(s)

J. Arbour

References

Arbour, J. In Prep. GoodFibes: an R package for the detection of muscle fibers from diceCT scans.

See Also

`sequencePlot`

Examples

```
images <- dir(system.file("extdata", package = "GoodFibes"), ".png", full.names = TRUE)

thresholdPlot(images, 1, 0.3)
thresholdPlot(images, 1, 0.4)
thresholdPlot(images, 1, 0.5)
```

Index

∗ **~internal**
    `pointsGenerator`, 23
∗ **datasets**
    `ant.final`, 3
    `ant.raw`, 4
∗ **package**
    `GoodFibes-package`, 2

`ant.final`, 3
`ant.raw`, 4

`check.overlap`, 5, *15*, *24*
`color.scale`, 6
`crop.stack`, 7, *9*

`equalize.stack`, *8*, 8

`fiber.angle`, 9, *11*, *13*
`fiber.curve`, 11, *13*
`fiber.lengths`, *10*, *11*, 12
`fibers.smoothed`, *6*, 13, *18*
`fuse.fibers`, *6*, 14, *15*, *24*
`fuse.fibers.auto`, 15

`good.fibes`, *5*, *6*, *13*, 16, *22*
`GoodFibes` *(*`GoodFibes-package`*)*, 2
`GoodFibes-package`, 2

`muscle.plot`, 19, *21*, *22*
`muscle.plot.multi`, *19*, 20, *22*
`muscle.plot.stl`, *7*, *19*, *21*, 21

`pointsGenerator`, 23

`quality.check`, *6*, *15*, 23

`sequencePlot`, 25, *27*

`thresholdPlot`, *16*, *26*, 26