Package 'BoneProfileR'

July 8, 2025

Type Package

Title Tools to Study Bone Compactness

Version 4.0

Date 2025-07-07

Depends imager, HelpersMG (>= 6.4), knitr, rmarkdown, R (>= 4.1), shiny

Suggests tiff, ijtiff, openxlsx, parallel, scatterplot3d, spatstat.geom, sp, fields, optimParallel

Description Bone Profiler is a scientific method and a software used to model bone section for paleontological and ecological studies. See Girondot and Laurin (2003) <`https://www.researchgate.net/publication/280021178_Bone_profiler_A_tool_to_quantify_model_and_statistically_compare_bone-section_compactness_profiles`> and Gônet, Laurin and Girondot (2022) <`https://palaeo-electronica.org/content/2022/3590-bone-section-compactness-model`>.

LazyData yes

LazyLoad yes

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Imports Rdpack

RdMacros Rdpack

Author Marc Girondot [aut, cre] (ORCID: <`https://orcid.org/0000-0001-6645-8530`>)

Maintainer Marc Girondot <`marc.girondot@gmail.com`>

Repository CRAN

Date/Publication 2025-07-07 22:00:06 UTC

Contents

`BoneProfileR-package` *A Model for Bone Compactness.*

Description

A Model for Bone Compactness.
The lastest version of this package can always been installed using:
install.packages(c("imager", "tiff", "ijtiff", "HelpersMG", "knitr", "rmarkdown", "openxlsx", "shiny", "fields"))
install.packages("http://marc.girondot.free.fr/CRAN/HelpersMG.tar.gz", repos=NULL, type="source")
install.packages("http://marc.girondot.free.fr/CRAN/BoneProfileR.tar.gz", repos=NULL, type="source")
BoneProfileR uses a new results management software that is developed as part of the HelpersMG package. Using this results management system (RM), all the results are stored as part of the analyzed image.
Online simplified version can be found at:
http://marc.girondot.free.fr/BoneProfilerR.html
This results management software has been developed to help users to maintain the results associated with the methodology used to obtain it. It is part of the large movement in science of replicative

research.
An analysis is then stored with the image in a single file with the following information:
name, timestamp, bg, fg, threshold, contour, centers, peripherie, compactness, array.compactness, cut.distance.center, cut.angle, used.centers, compactness.synthesis, partial, rotation.angle, global.compactness, optim, optimRadial
Several analyses can be stored within a single file.

Details

A model for bone compactness.

Package:	BoneProfileR
Type:	Package
Version:	4.0 build 1089
Date:	2025-07-07
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

References

Girondot M, Laurin M (2003) Bone Profiler: a tool to quantify, model, and statistically compare bone-section compactness profiles. Journal of Vertebrate Paleontology 23: 458-461

Laurin M, Girondot M, Loth M-M (2004) The evolution of long bone microstructure and lifestyle in lissamphibians. Paleobiology 30: 589-613

Gônet, Jordan, Michel Laurin, and Marc Girondot. 2022. BoneProfileR: The Next Step to Quantify, Model and Statistically Compare Bone Section Compactness Profiles. Paleontologica Electronica. 25(1): a12

Gônet, J., Bardin, J., Girondot, M., Hutchinson, J., Laurin, M., (2023). Deciphering locomotion in reptiles: application of elliptic Fourier transforms to femoral microanatomy. Zoological Journal of the Linnean Society 198, 1070-1091.

Gônet, J., Bardin, J., Girondot, M., Hutchinson, J.R., Laurin, M., (2023). Locomotor and postural diversity among reptiles viewed through the prism of femoral microanatomy: palaeobiological implications for some Permian and Mesozoic taxa. Journal of Anatomy 242, 891-916.

Gônet, J., Bardin, J., Girondot, M., Hutchinson, J.R., Laurin, M., (2023). Unravelling the postural diversity of mammals: contribution of humeral cross-sections to palaeobiological inferences. Journal of Mamalian Evolution 30, 321-337.

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 plot(bone, type="original")
 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
 bone <- BP_DetectCenters(bone=bone, analysis="logistic", method="Accurate")
 # Note that some parts of the section are concave but it does not give problems in the analysis
 # For section with very strong concavity, it is safer to use:
 # bone <- BP_DetectCenters(bone=bone, analysis="logistic", method="AccurateConvex")
 plot(bone, type="original")
 plot(bone, type="mineralized")
 plot(bone, type="unmineralized")
 plot(bone, type="section")
 plot(bone, type="colors")
 plot(bone, type="3Dcolors")
 bone <- BP_EstimateCompactness(bone, analysis="logistic", center="ontogenetic")
 plot(bone, type="original")
 plot(bone, type="mineralized")
 plot(bone, type="observations")
 bone <- BP_FitMLCompactness(bone, analysis="logistic")
 plot(bone, type="model", analysis=1)
```

```
 plot(bone, type="observations+model", analysis=1)
 fittedpar <- BP_GetFittedParameters(bone, analysis="logistic", type="global")[, "mean"]
 bone <- BP_DuplicateAnalysis(bone, from="logistic", to="flexit")
 bone <- BP_FitMLCompactness(bone,
                fitted.parameters=c(fittedpar, K1=1, K2=1),
                fixed.parameters=NULL, analysis="flexit")
 compare_AIC(Logistic=BP_GetFittedParameters(bone, analysis="logistic", ML=TRUE, return.all=TRUE),
         Flexit=BP_GetFittedParameters(bone, analysis="flexit", ML=TRUE, return.all=TRUE))
# pdf(file = "Figure 2.pdf", width = 8, height = 10, pointsize = 12)
layout(1:2)
plot(bone, type="observations+model", analysis="logistic", restorePar=FALSE, mar=c(4, 4, 2, 5))
plot(bone, type="observations+model", analysis="flexit", restorePar=FALSE, mar=c(4, 4, 2, 5))
layout(1)
# dev.off()
 out4p <- plot(bone, type="observations+model", analysis="logistic")
 out6p <- plot(bone, type="observations+model", analysis="flexit")
 bone <- BP_FitBayesianCompactness(bone, analysis="logistic")
 plot(bone, type="observations+model", CI="MCMC")
 bone <- BP_FitBayesianCompactness(bone, analysis="flexit")
 plot(bone, type="observations+model", CI="MCMC", analysis="flexit")
 plot(bone, type="mcmc", parameter="P",
      options.mcmc=list(xlim=c(0.55, 0.57), breaks=seq(from=0, to=1, by=0.001)))
 plot(bone, type="mcmc", parameter="S",
      options.mcmc=list(xlim=c(0.02, 0.05), breaks=seq(from=0.02, to=.05, by=0.001)))
 plot(bone, type="mcmc", parameter="Min",
      options.mcmc=list(xlim=c(0.05, 0.08), breaks=seq(from=0, to=1, by=0.001)))
 plot(bone, type="mcmc", parameter="Max",
      options.mcmc=list(xlim=c(0.95, 0.97), breaks=seq(from=0, to=1, by=0.001)))
 outMCMC <- RM_get(x = bone, RM = "RM", RMname = "logistic", valuename = "mcmc")
 summary(outMCMC)
 outMCMC <- RM_get(x = bone, RM = "RM", RMname = "flexit", valuename = "mcmc")
 summary(outMCMC)
 # pdf(file = "Figure 3.pdf", width = 8, height = 10, pointsize = 12)
 layout(1:2)
 plot(bone, type="mcmc", parameter="K1", analysis="flexit",
      options.mcmc=list(xlim=c(-1, 3), ylim=c(0,10),
      breaks=seq(from=-1, to=3, by=0.001),
      legend = FALSE, show.prior = FALSE, mar=c(4, 4, 1, 6)), restorePar=FALSE)
 segments(x0=1, x1=1,
         y0=0, y1=10, lty=4, lwd=3)
 text(x=ScalePreviousPlot(x=0.95, y=0.95)$x,
      y=ScalePreviousPlot(x=0.95, y=0.95)$y, labels="A", cex=3)
 plot(bone, type="mcmc", parameter="K2", analysis="flexit",
      options.mcmc=list(xlim=c(-1, 3), ylim=c(0,10),
      breaks=seq(from=-1, to=3, by=0.001),
      legend = FALSE, show.prior = FALSE, mar=c(4, 4, 1, 6)), restorePar=FALSE)
 segments(x0=1, x1=1,
         y0=0, y1=10, lty=4, lwd=3)
 text(x=ScalePreviousPlot(x=0.95, y=0.95)$x,
      y=ScalePreviousPlot(x=0.95, y=0.95)$y, labels="B", cex=3)
 # dev.off()

 bone <- BP_FitMLRadialCompactness(bone, analysis = "flexit")
```

```
 plot(bone, type="radial", parameter.name=c("P", "S"), analysis = "flexit")
 plot(bone, type="radial", parameter.name=c("P", "S", "Min", "Max"), analysis = "flexit")
 out <- RM_get(x=bone, RMname="flexit", valuename = "optimRadial")$synthesis
 mean(out[, "P"]); sd(out[, "P"])
 range(out[, "S"])
 quantile(out[, "S"])
 # pdf(file = "Figure 4.pdf", width=7, height = 9, pointsize = 12)
 layout(1:2)
 plot(bone, type="radial", parameter.name="P", analysis = "flexit", restorePar=FALSE)
 text(x=ScalePreviousPlot(x=0.95, y=0.95)$x,
      y=ScalePreviousPlot(x=0.95, y=0.95)$y, labels="A", cex=3)
 plot(bone, type="radial", parameter.name="S", analysis = "flexit", restorePar=FALSE)
 text(x=ScalePreviousPlot(x=0.95, y=0.95)$x,
      y=ScalePreviousPlot(x=0.95, y=0.95)$y, labels="B", cex=3)
 # dev.off()
 #' # How many times this package has been download
library(cranlogs)
BoneProfileR <- cran_downloads("BoneProfileR", from = "2021-10-07",
                            to = Sys.Date() - 1)
sum(BoneProfileR$count)
plot(BoneProfileR$date, BoneProfileR$count, type="l", bty="n",
     xlab="Download date", ylab="Number of downloads")

## End(Not run)
```

BP *Run a shiny application to fit bone section*

Description

Run a shiny application to fit bone section

Usage

```
BP()
```

Details

BP runs a shiny application to fit bone section

Value

Nothing

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
BP()

## End(Not run)
```

`BP_AutoFit` *Fit model automatically*

Description

Open an image, fit a model and generate a report.

Usage

```
BP_AutoFit(
  file = file.choose(),
  xlsx = TRUE,
  rotation.angle = 0,
  center = "ontogenetic"
)
```

Arguments

`file`	The file to be opened
`xlsx`	TRUE, FALSE or the name and path of the report
`rotation.angle`	The angle of rotation for analysis
`center`	Which center to be used.

Details

BP_AutoFit fits model automatically

Value

Characteristics of an image with all the fit information

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_AutoFit(file=path_Hedgehog, xlsx=TRUE)
 # or to open a dialog box
 bone <- BP_AutoFit()

## End(Not run)
```

`BP_ChooseBackground` *Let the use to choose the background color of an image*

Description

Let the user to choose the background color of an image.

Usage

```
BP_ChooseBackground(bone, analysis = 1)
```

Arguments

`bone`	The bone image to be used
`analysis`	The name or rank of analysis

Details

BP_ChooseBackground lets the use to choose the background color of an image

Value

The orignial bone object with a new attribute for background color

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
 path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.tif",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_ChooseBackground(bone=bone)
 bone <- BP_ChooseForeground(bone=bone)
 plot(bone)

## End(Not run)
```

`BP_ChooseCenter` *Let the user to choose the center of the bone*

Description

Let the user to choose the center of the bone.

Usage

```
BP_ChooseCenter(bone, analysis = 1)
```

Arguments

`bone`	The bone image to be used
`analysis`	The name or rank of analysis

Details

BP_ChooseCenter lets the use to choose the center of the bone

Value

The orignal bone object with a new attribute for center

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone)
 bone <- BP_DetectForeground(bone=bone)
 bone <- BP_ChooseCenter(bone=bone)
 # For partial section, only BP_ChooseCenter() must be used
 path_Dicynodon <- system.file("extdata", "Dicynodon_tibia_11.11.1.T_b_b-1.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Dicynodon)
 bone <- BP_DetectBackground(bone=bone)
 bone <- BP_DetectForeground(bone=bone)
 bone <- BP_ChooseCenter(bone=bone)
 bone <- BP_EstimateCompactness(bone, center="user", partial=TRUE)
 bone <- BP_FitMLCompactness(bone, analysis=1)
 plot(bone, type="observations+model")

## End(Not run)
```

`BP_ChooseForeground` *Let the user to choose the foreground color of an image*

Description

Let the user to choose the foreground color of an image.

Usage

```
BP_ChooseForeground(bone, analysis = 1)
```

Arguments

`bone`	The bone image to be used
`analysis`	The name or rank of analysis

Details

BP_ChooseForeground let the user to choose the foreground color of an image

Value

The orignial bone object with a new attribute for foreground color

Author(s)

Marc Girondot `<marc.girondot@u-psud.fr>`

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
 bone <- BP_OpenImage()
 bone <- BP_ChooseBackground(bone=bone)
 bone <- BP_ChooseForeground(bone=bone)
 plot(bone)

## End(Not run)
```

`BP_DetectBackground` *Detects the background color of an image*

Description

Detects the background color of an image.

Usage

```
BP_DetectBackground(bone, analysis = 1, show.plot = TRUE)
```

Arguments

`bone`	The bone image to be used
`analysis`	The name or rank of analysis
`show.plot`	should plot is shown ?

Details

BP_DetectBackground detects the background color of an image

Value

The orignial bone object with a new attribute for background color

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
 bone <- BP_OpenImage()
 bone <- BP_DetectBackground(bone=bone)
 bone <- BP_DetectForeground(bone=bone)
 plot(bone)

## End(Not run)
```

`BP_DetectCenters` *Detect the centers of an image*

Description

Detects the centers of an image. Note that this function must not be used with partial bone section. The method Fast works well with the convex bone section while if the section is concave, Accurate is slower but works well in all circonstances.
Fast method is maintained here only for compatibility with versions <3.1 of BoneProfileR.
If the section is concave, the methods FastConvex and AccurateConvex return a minimum convex section.

Usage

```
BP_DetectCenters(bone, analysis = 1, show.plot = TRUE, method = "Accurate")
```

Arguments

`bone`	The bone image to be used
`analysis`	The name or rank of analysis
`show.plot`	should plot is shown ?
`method`	Can be Fast, Accurate, FastConvex, or AccurateConvex

Details

BP_DetectCenters detects the centers of an image

Value

The orignial bone object with a new attribute for centers

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
 bone <- BP_OpenImage()
 # or
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone)
 bone <- BP_DetectForeground(bone=bone)
 bone <- BP_DetectCenters(bone=bone)
 plot(bone, type="mineralized", show.grid=FALSE)
 plot(bone, type="unmineralized", show.grid=FALSE)
 plot(bone, type="section", show.grid=FALSE)
 # Note that some parts of the section are concave but it does not give problems in the analysis
 # For section with very strong concavity, it could be safer to use:
 bone <- BP_DetectCenters(bone=bone, analysis="logistic", method="AccurateConvex")
 plot(bone, type="mineralized", show.grid=FALSE)
 plot(bone, type="unmineralized", show.grid=FALSE)
 plot(bone, type="section", show.grid=FALSE)
```

```
## End(Not run)
```

`BP_DetectForeground` *Detects the foreground color of an image*

Description

Detects the foreground color of an image.

Usage

```
BP_DetectForeground(bone, analysis = 1, show.plot = TRUE)
```

Arguments

`bone`	The bone image to be used
`analysis`	The name or rank of analysis
`show.plot`	should plot is shown ?

Details

BP_DetectForeground detects the foreground color of an image

Value

The orignial bone object with a new attribute for foreground color

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
 bone <- BP_OpenImage()
 bone <- BP_DetectBackground(bone=bone)
 bone <- BP_DetectForeground(bone=bone)
 plot(bone)

## End(Not run)
```

`BP_DuplicateAnalysis` *Duplicates an analysis stored in an object*

Description

Duplicates an analysis stored in an object.

Usage

```
BP_DuplicateAnalysis(bone, from = 1, to = 2)
```

Arguments

`bone`	The bone image to be used
`from`	The name or rank of analysis to be duplicated
`to`	The name or rank of analysis to be created

Details

BP_DuplicateAnalysis duplicates an analysis stored in an object

Value

The orignial bone object with a new analysis

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
```

```
 bone <- BP_DetectCenters(bone=bone, analysis="logistic")
 bone <- BP_EstimateCompactness(bone, analysis="logistic")
 bone <- BP_FitMLCompactness(bone, analysis="logistic")
 plot(bone)
 plot(bone, type="observations")
 plot(bone, type="observations+model", analysis=1)
 bone <- BP_DuplicateAnalysis(bone, from="logistic", to="flexit")

## End(Not run)
```

`BP_EstimateCompactness`

Estimation of the compactness of a bone section

Description

Estimation of the compactness of a bone section.
The reference for radial estimation of compactness is the trigonometric circle for rotation.angle=0 in BP_EstimateCompactness():

- The top of the section is located at -pi/2.
- The left of the section is located at -pi and +pi.
- The bottom of the section is located at pi/2.
- The right of the section is 0.
  If rotation.angle is different from 0, the value of rotation.angle is added to the angle modulo 2.pi.
  The method Fast works well with the convex bone section while if the section is concave, Accurate is slower but works well in all circonstances.
  Fast method is maintained here only for compatibility with versions <3.1 of BoneProfileR.
  If the section is concave, the methods FastConvex and AccurateConvex return a minimum convex section.
  If the center has been automatically detected, the method parameter is ignored because it has already been used with the function BP_DetectCenters().

Usage

```
BP_EstimateCompactness(
  bone,
  center = "ontogenetic",
  partial = FALSE,
  NbRemoveEdgePartial = 1,
  cut.angle = 60,
  cut.distance = 100,
```

```
  NbRemoveDistanceExterior = 1,
  rotation.angle = 0,
  analysis = 1,
  method = "Fast",
  show.plot = TRUE,
  cut.max = 360
)
```

Arguments

`bone`	The bone image to be used
`center`	Which center to be used: user, mineralized, unmineralized, section, ontogenetic
`partial`	Is the section partial?
`NbRemoveEdgePartial`	How many radial section to remove at the edge of partial section?
`cut.angle`	Number of angles
`cut.distance`	Number of distances
`NbRemoveDistanceExterior`	How many exterior sectors should be removed from analysis?
`rotation.angle`	The angle of rotation for analysis
`analysis`	The name or rank of analysis
`method`	Can be Fast, Accurate, FastConvex, or AccurateConvex
`show.plot`	should plot is shown ?
`cut.max`	The number of slices for the internal estimation

Details

BP_EstimateCompactness estimates the compactness of a bone section

Value

The orignial bone object with a new attribute for compactness

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
 bone <- BP_OpenImage()
 # or
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone)
 bone <- BP_DetectForeground(bone=bone)
 bone <- BP_DetectCenters(bone=bone)
 bone <- BP_EstimateCompactness(bone)
 plot(bone, type="original", show.grid=FALSE)
 plot(bone, type="mineralized", show.grid=FALSE)
 plot(bone, type="unmineralized", show.grid=FALSE)
 plot(bone, type="section", show.grid=FALSE)
 plot(bone, type="original", show.grid=TRUE)

## End(Not run)
```

`BP_FitBayesianCompactness` *Estimation of Bayesian model of a bone section*

Description

Estimation of Bayesian model of a bone section./r Get information using ?MHalgoGen.

Usage

```
BP_FitBayesianCompactness(
  bone = stop("A result from BP_FitMLCompactness() must be provided"),
  priors = NULL,
  n.iter = 10000,
  n.chains = 1,
  n.adapt = 5000,
  thin = 10,
  analysis = 1,
  adaptive = TRUE,
  silent = TRUE
)
```

Arguments

`bone`	The bone image to be used
`priors`	Priors
`n.iter`	Number of iterations

`n.chains`	Number of chains
`n.adapt`	Number of iteration to adapt
`thin`	Thin parameter for analysis
`analysis`	Name or rank of analysis
`adaptive`	Should SDProp be changed during iterations
`silent`	Should some information must me shown ?

Details

BP_FitBayesianCompactness estimates Bayesian model of a bone section

Value

The -Ln L

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
 library(BoneProfileR)
 path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
 bone <- BP_DetectCenters(bone=bone, analysis="logistic")
 bone <- BP_EstimateCompactness(bone, analysis="logistic")
 bone <- BP_FitMLCompactness(bone, analysis="logistic")
 plot(bone)
 plot(bone, type="observations")
 plot(bone, type="observations+model", analysis=1)
 fittedpar <- BP_GetFittedParameters(bone, analysis="logistic", ML=TRUE, return.all=FALSE)[, "mean"]
 bone <- BP_DuplicateAnalysis(bone, from="logistic", to="flexit")
 bone <- BP_FitMLCompactness(bone,
                fitted.parameters=c(fittedpar, K1=1, K2=1),
                fixed.parameters=NULL, analysis="flexit")
 compare_AIC(Logistic=BP_GetFittedParameters(bone, analysis="logistic", ML=TRUE, return.all=TRUE),
```

```
        Flexit=BP_GetFittedParameters(bone, analysis="flexit", ML=TRUE, return.all=TRUE))
 out4p <- plot(bone, type="observations+model", analysis="logistic")
 out6p <- plot(bone, type="observations+model", analysis="flexit")
 bone <- BP_FitBayesianCompactness(bone, analysis="logistic")
 plot(bone, type="observations+model", CI="MCMC", analysis="logistic")
 bone <- BP_FitBayesianCompactness(bone, analysis="flexit")
 plot(bone, type="observations+model", CI="MCMC", analysis="flexit")

## End(Not run)
```

`BP_FitBayesianPeriodicCompactness`

Estimation of the likelihood of a bone section

Description

Estimation of the compactness of a bone section using Bayesian periodic model.
To control the parallel computing, use:
options(mc.cores = [put here the number of cores you want use])
options(forking = FALSE) or options(forking = TRUE)
The maximum number of cores is obtained by: parallel::detectCores()

Usage

```
BP_FitBayesianPeriodicCompactness(
  bone,
  fitted.parameters = NULL,
  priors = NULL,
  fixed.parameters = NULL,
  analysis = 1,
  silent = FALSE,
  replicates.CI = 2000,
  amplitude.max = 0.1,
 control.MHalgoGen = list(n.iter = 10000, n.chains = 1, trace = TRUE, n.adapt = 5000,
    thin = 1, adaptive = TRUE)
)
```

Arguments

`bone`	The bone image to be used
`fitted.parameters`	Parameters of the model to be fitted
`priors`	The priors of Bayesian analysis
`fixed.parameters`	Fixed parameters of the model
`analysis`	Name or rank of analysis

`silent`	Should the function displays some information?
`replicates.CI`	Number of replicates to estimate confidence interval using Hessian
`amplitude.max`	The maximum allowed amplitude for each parameter
`control.MHalgoGen`	The control parameters of MHalgoGen()

Details

BP_FitBayesianPeriodicCompactness estimates likelihood of global model of a bone section

Value

The -Ln L

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactne` `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
 bone <- BP_DetectCenters(bone=bone, analysis="logistic")
 bone <- BP_EstimateCompactness(bone, analysis="logistic", cut.angle = 60)
 bone <- BP_FitMLCompactness(bone, analysis="logistic", twosteps=TRUE)
 plot(bone, type="observations+model", analysis="logistic")
 par <- BP_GetFittedParameters(bone, analysis="logistic", ML=TRUE, return.all=FALSE)[, "mean"]
 options(mc.cores=parallel::detectCores())

 ##############################################
 # Periodic analysis
 ##############################################
 bone <- BP_FitMLPeriodicCompactness(bone, analysis="logistic", control.optim=list(trace=2),
                                     fitted.parameters=c(par, PSin=0.001, PCos=0.001,
                                     SSin=0.001, SCos=0.001, MinSin=0.001, MinCos=0.001,
                                     MaxSin=0.001, MaxCos=0.001), replicates.CI=2000)
 bone <- BP_FitBayesianPeriodicCompactness(bone, analysis="logistic", replicates.CI=2000)
```

```
 mcmc <- RM_get(bone, RMname="logistic", valuename="mcmcPeriodic")
 plot(mcmc, parameters="P", what="MarkovChain", ylim=c(0.555, 0.565), main="P parameter")

 plot(bone, type="mcmcPeriodic", parameter.name="compactness", col=rainbow(128))
 plot(bone, type="mcmcPeriodic", parameter.name="compactness",
              col=hcl.colors(12, "YlOrRd", rev = TRUE))
 plot(bone, type="mcmcPeriodic", parameter.name="averagemodel")
 plot(bone, type="mcmcPeriodic", parameter.name="P",
              rgb(red = 0.7, green = 0.7, blue = 0.7, alpha = 0.2))
 plot(bone, type="mcmcPeriodic", parameter.name="P", ylim=c(0, 1),
              rgb(red = 0.7, green = 0.7, blue = 0.7, alpha = 0.2))


## End(Not run)
```

`BP_FitMLCompactness` *Estimation of the likelihood of a bone section*

Description

Estimation of the model of compactness of a bone section.
The two-steps analysis performs first a quasi-Newton method, then a Bayesian MCMC and finally again a quasi-Newton method. It generally ensures that global minimum is found. On the other hand, it doubles the time to complete.

Usage

```
BP_FitMLCompactness(
  bone,
  fitted.parameters = c(P = 0.5, S = 0.02, Min = 0.001, Max = 0.999),
  priors = NULL,
  fixed.parameters = c(K1 = 1, K2 = 1),
  twosteps = TRUE,
  replicates.CI = 10000,
  analysis = 1,
  control.optim = list(trace = 1),
  silent = FALSE
)
```

Arguments

`bone`	The bone image to be used
`fitted.parameters`	Parameters of the model to be fitted
`priors`	Priors used for intermediate estimations
`fixed.parameters`	Fixed parameters of the model

`twosteps`	Does a 2-steps analysis be performed?
`replicates.CI`	Number of replicates to estimate confidence interval
`analysis`	Name or rank of analysis
`control.optim`	The list of options for optim.
`silent`	Should information be shown?

Details

BP_FitMLCompactness estimates likelihood of model of a bone section

Value

The -Ln L

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
 bone <- BP_OpenImage()
 # or, to use the package imager to open a tiff image
 bone <- BP_OpenImage(ijtiff=TRUE)
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
 bone <- BP_DetectCenters(bone=bone, analysis="logistic")
 bone <- BP_EstimateCompactness(bone, analysis="logistic")
 plot(bone, type="mineralized", show.grid=FALSE)
 plot(bone, type="unmineralized", show.grid=FALSE)
 plot(bone, type="section", show.grid=FALSE)
 bone <- BP_FitMLCompactness(bone, analysis="logistic", twosteps=TRUE)
 BP_GetFittedParameters(bone, analysis="logistic", ML=TRUE, return.all=FALSE)
 plot(bone)
 plot(bone, type="observations")
```

```
 plot(bone, type="observations+model", analysis=1)
 bone <- BP_DuplicateAnalysis(bone, from="logistic", to="flexit")
fittedpar <- BP_GetFittedParameters(bone, analysis="logistic", ML=TRUE, return.all=FALSE)[, "mean"]
 bone <- BP_DuplicateAnalysis(bone, from="logistic", to="flexit")
 BP_ListAnalyses(bone)
 bone <- BP_FitMLCompactness(bone,
                fitted.parameters=c(fittedpar, K1=1, K2=1),
                fixed.parameters=NULL, analysis="flexit", twosteps=TRUE)
compare_AIC(Logistic=BP_GetFittedParameters(bone, analysis="logistic", , ML=TRUE, return.all=TRUE),
        Flexit=BP_GetFittedParameters(bone, analysis="flexit", , ML=TRUE, return.all=TRUE))
 out4p <- plot(bone, type="observations+model", analysis="logistic")
 out6p <- plot(bone, type="observations+model", analysis="flexit")

## End(Not run)
```

BP_FitMLPeriodicCompactness

Estimation of the likelihood of a bone section

Description

Estimation of the compactness of a bone section using radial model.
If the fitted.parameters and fixed.parameters are NULL and the analysis includes a BP_FitMLCompactness() result, the values of this result is used as a reference for fitted.parameters and fixed.parameters.
If no BP_FitMLCompactness() result is available, it will use:
fitted.parameters=c(P=0.5, S=0.05, Min=-2, Max=5); fixed.parameters=c(K1=1, K2=1).
The reference for radial estimation of compactness is the trigonometric circle for rotation.angle=0 in BP_EstimateCompactness():

- The top of the section is located at -pi/2.
- The left of the section is located at -pi and +pi.
- The bottom of the section is located at pi/2.
- The right of the section is 0.
  If rotation.angle is different from 0, the value of rotation.angle is added to the angle modulo 2.pi.
  The two-steps analysis performs first a quasi-Newton method, then a Bayesian MCMC and finally again a quasi-Newton method. It generally ensures that global minimum is found. On the other hand, it doubles the time to complete for each angle.
  To control the parallel computing, use:
  options(mc.cores = [put here the number of cores you want use])
  options(forking = FALSE) or options(forking = TRUE)
  The maximum number of cores is obtained by: parallel::detectCores()

Usage

```
BP_FitMLPeriodicCompactness(
  bone,
  fitted.parameters = NULL,
  fixed.parameters = NULL,
  analysis = 1,
  silent = FALSE,
  replicates.CI = NULL,
  twosteps = FALSE,
  amplitude.max = 0.1,
  control.optim = list(trace = 1)
)
```

Arguments

`bone`	The bone image to be used
`fitted.parameters`	Parameters of the model to be fitted
`fixed.parameters`	Fixed parameters of the model
`analysis`	Name or rank of analysis
`silent`	Should the function displays some information?
`replicates.CI`	Number of replicates to estimate confidence interval using Hessian
`twosteps`	Should a 2-steps analysis be performed? It can be sometimes useful.
`amplitude.max`	The maximum allowed amplitude for each parameter
`control.optim`	The list of options for optim.

Details

BP_FitMLPeriodicCompactness estimates likelihood of global model of a bone section

Value

The -Ln L

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
 bone <- BP_DetectCenters(bone=bone, analysis="logistic")
 bone <- BP_EstimateCompactness(bone, analysis="logistic", cut.angle = 60)
 bone <- BP_FitMLCompactness(bone, analysis="logistic", twosteps=TRUE)
 plot(bone, type="observations+model", analysis="logistic")
 par <- BP_GetFittedParameters(bone, analysis="logistic", ML=TRUE, return.all=FALSE)[, "mean"]
 options(mc.cores=parallel::detectCores())

 #############################################
 # Periodic analysis
 #############################################
 bone <- BP_FitMLPeriodicCompactness(bone, analysis="logistic", control.optim=list(trace=2),
                                     fitted.parameters=c(par, PSin=0.001, PCos=0.001,
                                     SSin=0.001, SCos=0.001, MinSin=0.001, MinCos=0.001,
                                     MaxSin=0.001, MaxCos=0.001), replicates.CI=2000)
 analysisP <- BP_GetFittedParameters(bone, analysis="logistic", type="periodic",
                                     ML=TRUE, return.all=FALSE)[, "mean"]
 analysisP$par
 plot(bone, type="periodic", parameter.name="compactness", col=rainbow(128))
 plot(bone, type="periodic", parameter.name="compactness",
              col=hcl.colors(12, "YlOrRd", rev = TRUE))
 plot(bone, type="periodic", parameter.name="averagemodel")
 plot(bone, type="periodic", parameter.name="P",
              rgb(red = 0.7, green = 0.7, blue = 0.7, alpha = 0.2))
 plot(bone, type="periodic", parameter.name="P", ylim=c(0, 1),
              rgb(red = 0.7, green = 0.7, blue = 0.7, alpha = 0.2))
 boneNoPeriodic <- BP_FitMLPeriodicCompactness(bone, analysis="logistic",
                                               control.optim=list(trace=2),
                                     fitted.parameters=par, replicates.CI=2000)
 analysisNP <- BP_GetFittedParameters(boneNoPeriodic, analysis="logistic", ML=TRUE,
                                      return.all=TRUE, type="periodic")
 analysisNP$par
 compare_AIC(PeriodicModel=analysisP,
             NoPeriodicModel=analysisNP)

 #############################################

 # Note that the absolute likelihood is dependent on the number of angle cut
 # Only models analyzed with the same number of angle cuts can be compared

 dbinom(5, 10, prob=0.4, log=TRUE);
       dbinom(2, 5, prob=0.4, log=TRUE)+dbinom(3, 5, prob=0.4, log=TRUE)
 # But the likelihood difference between two models are not:
 dbinom(5, 10, prob=0.4, log=TRUE)-dbinom(5, 10, prob=0.3, log=TRUE)
```

```
 dbinom(2, 5, prob=0.4, log=TRUE)+dbinom(3, 5, prob=0.4, log=TRUE)-
      dbinom(2, 5, prob=0.3, log=TRUE)-dbinom(3, 5, prob=0.3, log=TRUE)

## End(Not run)
```

BP_FitMLRadialCompactness

Estimation of the likelihood of a bone section

Description

Estimation of the compactness of a bone section using radial model.
If the fitted.parameters and fixed.parameters are NULL and the analysis includes a BP_FitMLCompactness() result, the values of this result is used as a reference for fitted.parameters and fixed.parameters.
If no BP_FitMLCompactness() result is available, it will use:
fitted.parameters=c(P=0.5, S=0.05, Min=-2, Max=5); fixed.parameters=c(K1=1, K2=1).
The reference for radial estimation of compactness is the trigonometric circle for rotation.angle=0 in BP_EstimateCompactness():

- The top of the section is located at -pi/2.
- The left of the section is located at -pi and +pi.
- The bottom of the section is located at pi/2.
- The right of the section is 0.
  If rotation.angle is different from 0, the value of rotation.angle is added to the angle modulo 2.pi.
  The two-steps analysis performs first a quasi-Newton method, then a Bayesian MCMC and finally again a quasi-Newton method. It generally ensures that global minimum is found. On the other hand, it doubles the time to complete for each angle.
  To control the parallel computing, use:
  options(mc.cores = [put here the number of cores you want use])
  options(forking = FALSE) or options(forking = TRUE)
  The maximum number of cores is obtained by: parallel::detectCores()

Usage

```
BP_FitMLRadialCompactness(
  bone,
  fitted.parameters = NULL,
  priors = NULL,
  fixed.parameters = NULL,
  analysis = 1,
```

```
  silent = FALSE,
  twosteps = TRUE,
  progressbar = FALSE
)
```

Arguments

`bone`	The bone image to be used
`fitted.parameters`	Parameters of the model to be fitted
`priors`	If twosteps is TRUE, tell what prior should be used.
`fixed.parameters`	Fixed parameters of the model
`analysis`	Name or rank of analysis
`silent`	Should the function displays some information?
`twosteps`	Should a 2-steps analysis be performed?
`progressbar`	Should a progress bar be shown?

Details

BP_FitMLRadialCompactness estimates likelihood of model of a bone section

Value

The -Ln L

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 # or
```

```
bone <- BP_OpenImage(ijtiff=TRUE)
bone <- BP_DetectBackground(bone=bone, analysis="logistic")
bone <- BP_DetectForeground(bone=bone, analysis="logistic")
bone <- BP_DetectCenters(bone=bone, analysis="logistic")
bone <- BP_EstimateCompactness(bone, analysis="logistic")
bone <- BP_EstimateCompactness(bone, analysis="logistic", cut.angle=30)
bone <- BP_FitMLCompactness(bone, analysis="logistic")
plot(bone)
plot(bone, type="observations")
plot(bone, type="observations+model", analysis=1)
fittedpar <- BP_GetFittedParameters(bone, analysis="logistic",
                                    ML=TRUE, return.all = FALSE)[, "mean"]
bone <- BP_DuplicateAnalysis(bone, from="logistic", to="flexit")
bone <- BP_FitMLCompactness(bone,
               fitted.parameters=c(fittedpar, K1=1.01, K2=1.01),
               fixed.parameters=NULL, analysis="flexit")
bone <- BP_FitBayesianCompactness(bone, analysis="flexit")
mcmc <- RM_get(bone, RMname = "flexit", value="mcmc")
fittedpar <- as.parameters(mcmc)
bone <- BP_FitMLCompactness(bone,
               fitted.parameters=fittedpar,
               fixed.parameters=NULL, analysis="flexit")
compare_AIC(Logistic=BP_GetFittedParameters(bone, analysis="logistic",
                                            ML=TRUE, return.all = TRUE),
            Flexit=BP_GetFittedParameters(bone, analysis="flexit",
                                          ML=TRUE, return.all=TRUE))
out4p <- plot(bone, type="observations+model", analysis="logistic")
out6p <- plot(bone, type="observations+model", analysis="flexit")
# The twosteps fit is more acurate but is around 100 times slower
bone <- BP_FitMLRadialCompactness(bone, analysis="logistic", twosteps=TRUE)
bone <- BP_FitMLRadialCompactness(bone, analysis="logistic", twosteps=FALSE)
plot(bone, type="observations", angle=0)
plot(bone, type="model", analysis="logistic", angle=0)
plot(bone, type="observations+model", angle=0)
plot(bone, type="observations+model", angle=pi)
plot(bone, type="radial", parameter.name=c("P"), analysis="logistic")
plot(bone, type="radial", parameter.name=c("P", "S"), analysis="logistic")
plot(bone, type="radial", parameter.name=c("P", "S", "Min", "Max"), analysis="logistic")
plot(bone, type="radial", parameter.name=c("TRC"), analysis="logistic")

# The observed compactness
plot(bone, type="radial", parameter.name=c("observed.compactness"), analysis="logistic")
# The observed compactness weighted by the pixel number
plot(bone, type="radial", parameter.name="linearized.observed.compactness", analysis="logistic")
# The integration of the compactness model
plot(bone, type="radial", parameter.name="modeled.compactness", analysis="logistic")
# The integration of the compactness model weighted by the pixel number
plot(bone, type="radial", parameter.name="linearized.modeled.compactness", analysis="logistic")

# Test using the change of orientation using default.angle from BP_EstimateCompactness():
bone <- BP_DuplicateAnalysis(bone, from="logistic", to="logistic_rotation_pi")
# With a pi rotation, the top moves to the bottom and the left moves to the right
bone <- BP_EstimateCompactness(bone, rotation.angle=pi, analysis="logistic_rotation_pi")
```

```
bone <- BP_FitMLRadialCompactness(bone, analysis="logistic_rotation_pi")
plot(bone, type="radial", parameter.name=c("P", "S"), analysis="logistic")
plot(bone, type="radial", parameter.name=c("P", "S"), analysis="logistic_rotation_pi")
BP_Report(bone=bone,
          analysis=1,
          docx=NULL,
          pdf=NULL,
          xlsx=file.path(getwd(), "report.xlsx"),
          author="Marc Girondot",
          title=attributes(bone)$name)

## End(Not run)
```

`BP_flexit` *Return the flexit for BoneProfileR models*

Description

Return a vector with the probabilities. The flexit equation is published in:
Abreu-Grobois, F.A., Morales-Mérida, B.A., Hart, C.E., Guillon, J.-M., Godfrey, M.H., Navarro, E. & Girondot, M. (2020) Recent advances on the estimation of the thermal reaction norm for sex ratios. PeerJ, 8, e8451.
If dose < P then $(1 + (2^K1 - 1) * exp(4 * S1 * (P - x)))^(-1/K1)$
If dose > P then $1 - ((1 + (2^K2 - 1) * exp(4 * S2 * (x - P)))^(-1/K2)$
with:

$$S1 = (2^(K1 - 1) * S * K1)/(2^K1 - 1)$$

$$S2 = (2^(K2 - 1) * S * K2)/(2^K2 - 1)$$

New in version 4.7-3 and larger:

If 2^K1 is too large to be estimated, the approximation $S1 = S * K1/2$ is used.
Demonstration:

$$S1 = (2^(K1 - 1) * S * K1)/(2^K1 - 1)$$

$$S1 = exp(log((2^(K1 - 1) * S * K1)/(2^K1 - 1)))$$

$$S1 = exp(log(2^(K1 - 1)) + log(S * K1) - log(2^K1 - 1))$$

When $K1$ is very large, $2^K1 - 1 = 2^K1$ then

$$S1 = exp((K1 - 1) * log(2) + log(S * K1) - K1 * log(2))$$

$$S1 = exp((K1 * log(2) - log(2) + log(S * K1) - K1 * log(2))$$

$$S1 = exp(log(S * K1) - log(2))$$

$$S1 = S * K1/2$$

If 2^K2 is too large to be estimated, the approximation $S2 = S * K2/2$ is used.

If $(1+(2^K1-1)*exp(4*S1*(P-x)))^(-1/K1)$ is not finite, the following approximation is used:

$$exp((-1/K1)*(K1*log(2)+(4*S1*(P-x))))$$

If $1-((1+(2^K2-1)*exp(4*S2*(x-P)))^(-1/K2)$ is not finite, the following approximation is used:

$$1-exp((-1/K2)*(K2*log(2)+(4*S2*(x-P))))$$

Usage

```
BP_flexit(
  x,
  par = NULL,
  P = NULL,
  S = NULL,
  K1 = NULL,
  K2 = NULL,
  Min = 0,
  Max = 1,
  zero = 1e-10,
  error0 = 1e-10,
  error1 = 1 - 1e-10
)
```

Arguments

`x`	The values at which the flexit model must be calculated
`par`	The vector with P, S, K1, and K2 values
`P`	P value
`S`	S value
`K1`	K1 value
`K2`	K2 value
`Min`	Min value for scaled flexit model
`Max`	Max value for scaled flexit model
`zero`	Value to replace zero
`error0`	Value to return if an error is observed toward 0
`error1`	Value to return if an error is observed toward 1

Details

Return the flexit value

Value

A vector with the probabilities

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

Examples

```
n <- flexit(x=1:100, par=c(P=50, S=0.001, K1=0.01, K2=0.02))
n <- flexit(x=1:100, P=50, S=0.001, K1=0.01, K2=0.02)

1/(1+exp(0.01*4*(50-1:100)))
flexit(1:100, P=50, S=0.01, K1=1, K2=1)
```

`BP_GetFittedParameters` *Return the fitted parameters*

Description

Return the fitted parameters or complete object.

Usage

```
BP_GetFittedParameters(
  bone,
  analysis = 1,
  return.all = FALSE,
  ML = TRUE,
  periodic = FALSE,
  type = "global"
)
```

Arguments

`bone`	The bone image to be used
`analysis`	Name or rank of analysis
`return.all`	If TRUE, return the complete object
`ML`	If TRUE, return the ML estimate and the SE ; if FALSE, returns the MCMC estimate
`periodic`	If TRUE, the periodic model is used (depreciated, use type)
`type`	Can be "global", "radial", or "periodic"

Details

BP_GetFittedParameters returns the fitted parameters

Value

The fitted parameters

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
 bone <- BP_DetectCenters(bone=bone, analysis="logistic")
 bone <- BP_EstimateCompactness(bone, analysis="logistic")
 bone <- BP_FitMLCompactness(bone, analysis="logistic")
 BP_GetFittedParameters(bone, analysis="logistic")

## End(Not run)
```

`BP_ListAnalyses` *List the analyses stored in an object*

Description

Get the analyses stored in an object.

Usage

```
BP_ListAnalyses(bone, silent = TRUE, max.level = FALSE)
```

Arguments

`bone`	The bone image to be used
`silent`	Should the results be shown ?
`max.level`	If TRUE, will return all list element of the objects

Details

BP_ListAnalyses lists the analyses stored in an object

Value

The list of analyses

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
 bone <- BP_DetectCenters(bone=bone, analysis="logistic")
 bone <- BP_EstimateCompactness(bone, analysis="logistic")
 bone <- BP_FitMLCompactness(bone, analysis="logistic")
 plot(bone)
 plot(bone, type="observations")
 plot(bone, type="observations+model", analysis=1)
 bone <- BP_DuplicateAnalysis(bone, from="logistic", to="flexit")
 BP_ListAnalyses(bone)

## End(Not run)
```

`BP_LnLCompactness` *Estimation of the likelihood of a bone section*

Description

Estimation of the compactness of a bone section.

Usage

```
BP_LnLCompactness(
  par,
  bone = NULL,
  data_m = NULL,
  data_nm = NULL,
  distance.center = NULL,
  fixed.parameters = NULL,
  analysis = 1,
  sign = -1
)
```

Arguments

`par`	Parameters of the model
`bone`	The bone image to be used
`data_m`	Number of mineralized pixels
`data_nm`	Number of non-mineralized pixels
`distance.center`	Distances to the center
`fixed.parameters`	Fixed parameters of the model
`analysis`	Name or rank of analysis
`sign`	The likelihood if multiplied by sign (-1 or +1) to return -Ln L or Ln L

Details

BP_LnLCompactness estimates likelihood of model of a bone section

Value

The -Ln L

Author(s)

Marc Girondot `<marc.girondot@gmail.com>`

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone)
 bone <- BP_DetectForeground(bone=bone)
 bone <- BP_DetectCenters(bone=bone)
 bone <- BP_EstimateCompactness(bone)
 plot(bone)

## End(Not run)
```

`BP_OpenImage` *Open an image*

Description

Open an image.

Usage

```
BP_OpenImage(file = file.choose(), name = NULL, ijtiff = FALSE)
```

Arguments

`file`	The file to be opened
`name`	Name of this slice
`ijtiff`	Should the ijtiff must be used to read tiff image

Details

BP_OpenImage opens an image

Value

Characteristics of an image

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 plot(bone)
 path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.tif",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 plot(bone)
 bone <- BP_OpenImage(file=path_Hedgehog, ijtiff=TRUE)
 plot(bone)
 # A partial section
 path_Dicynodon <- system.file("extdata", "Dicynodon_tibia_11.11.1.T_b_b-1.png",
                             package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Dicynodon)
 plot(bone)
 # To open a file with a dialog:
 bone <- BP_OpenImage()

## End(Not run)
```

`BP_Report` *Generate a pdf report for the analyzed bone*

Description

Generate a docx, xlsx, or pdf report.
In the xlsx report, the observed compactness is simply the ratio of the number of mineralized pixels to the number of total pixels.
The corrected modelled compactness is the compactness if the bone had the same number of pixels in the centre as on the periphery So it's a compactness that corrects for the shape of the bone.

Usage

```
BP_Report(
  bone = stop("A bone section must be provided"),
```

```
  control.plot = list(message = NULL, show.centers = TRUE, show.colors = TRUE, show.grid
    = TRUE, CI = "ML", show.legend = TRUE),
  analysis = 1,
  docx = file.path(getwd(), "report.docx"),
  pdf = file.path(getwd(), "report.pdf"),
  xlsx = file.path(getwd(), "report.xlsx"),
  author = NULL,
  title = attributes(bone)$name
)
```

Arguments

`bone`	The bone image
`control.plot`	A list with the parameters used for plot
`analysis`	Indicate analysis name or rank that you want report
`docx`	Name of Word file
`pdf`	Name of pdf file
`xlsx`	Name of Excel file
`author`	Name indicated in the report
`title`	Title of the report

Details

BP_Report save a pdf report for the analyzed bone

Value

Nothing

Author(s)

Marc Girondot `<marc.girondot@gmail.com>`

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
```

```
                          package = "BoneProfileR")
bone <- BP_OpenImage(file=path_Hedgehog)
bone <- BP_DetectBackground(bone=bone, analysis="logistic")
bone <- BP_DetectForeground(bone=bone, analysis="logistic")
bone <- BP_DetectCenters(bone=bone, analysis="logistic")
bone <- BP_EstimateCompactness(bone, analysis="logistic")
bone <- BP_FitMLCompactness(bone, analysis="logistic")
fittedpar <- BP_GetFittedParameters(bone, analysis="logistic",
                                    ML=TRUE, return.all = FALSE)[, "mean"]
bone <- BP_DuplicateAnalysis(bone, from="logistic", to="flexit")
bone <- BP_FitMLCompactness(bone,
              fitted.parameters=c(fittedpar, K1=1, K2=1),
              fixed.parameters=NULL, analysis="flexit")
compare_AIC(Logistic=BP_GetFittedParameters(bone, analysis="logistic",
                                            ML=TRUE, return.all = TRUE),
            Flexit=BP_GetFittedParameters(bone, analysis="flexit",
                                          ML=TRUE, return.all = TRUE))
bone <- BP_FitMLRadialCompactness(bone, analysis="logistic")
# Test using the change of orientation using default.angle from BP_EstimateCompactness():
bone <- BP_DuplicateAnalysis(bone, from="logistic", to="logistic_rotation_pi")
# With a pi rotation, the top moves to the bottom and the left moves to the right
bone <- BP_EstimateCompactness(bone, rotation.angle=pi, analysis="logistic_rotation_pi")
bone <- BP_FitMLRadialCompactness(bone, analysis="logistic_rotation_pi")
# Periodic analysis
par <- BP_GetFittedParameters(bone, analysis="logistic", ML=TRUE, return.all = FALSE)[, "mean"]
options(mc.cores=parallel::detectCores())
bone <- BP_FitMLPeriodicCompactness(bone, analysis="logistic", control.optim=list(trace=2),
                                    fitted.parameters=c(par, PSin=0.001, PCos=0.001,
                                    SSin=0.001, SCos=0.001, MinSin=0.001, MinCos=0.001,
                                    MaxSin=0.001, MaxCos=0.001), replicates.CI=2000)

BP_Report(bone=bone,
          analysis=1,
          docx=NULL,
          pdf=NULL,
          xlsx=file.path(getwd(), "report.xlsx"),
          author="Marc Girondot",
          title=attributes(bone)$name)

BP_Report(bone=bone,
          analysis=1,
          docx=NULL,
          pdf=file.path(getwd(), "report.pdf"),
          xlsx=NULL,
          author="Marc Girondot",
          title=attributes(bone)$name)

BP_Report(bone=bone,
          analysis=1,
          docx=file.path(getwd(), "report.docx"),
          pdf=NULL,
          xlsx=NULL,
          author="Marc Girondot",
```

```
            title=attributes(bone)$name)

## End(Not run)
```

`Erinaceus_europaeus` *Example of hedgehog femur*

Description

Example of hedgehog femur. A drawing produced by Michel Laurin.

Usage

```
Erinaceus_europaeus
```

Format

A png image.

Details

Example of hedgehog femur

Author(s)

Marc Girondot `<marc.girondot@gmail.com>`

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `plot.BoneProfileR()`, `summary.BoneProfileR()`

Examples

```
bone <- Erinaceus_europaeus
plot(bone)
```

`plot.BoneProfileR` *Plot a bone section*

Description

Display a bone section.
type value can be:
Image plot: `original`, `mineralized`, `unmineralized`, `section`
Original is the original image, mineralized is the mineral interpretation of the section, unmineralized is the unmineralized interpretation of the section, section is the interpretation of the section.
`colors` shows the histograms of pixel information with foreground and background colors if they are defined.
`3Dcolors` show the pixels colors in 3D
Global analysis: `observations`, `model`, `observations+model`
Radial analysis: `radial`
Periodic analysis: `periodic`
If angle is not null and a radial analysis exists, it will show the model for this angle.
`mcmc`: It will show the posterior distribution of parameter.
For periodic analysis, you can see a particular parameter with parameter.name being P, S, Min, Max, K1, or K2 or the global median compactness using parameter.name="compactness". You can use col=rainbow(128) or hcl.colors(128) to see the region of transition. You can also plot the average compactness using parameter.name="averagemodel".

Usage

```
## S3 method for class 'BoneProfileR'
plot(
  x,
  message = NULL,
  type = "original",
  angle = NULL,
  show.all.angles = FALSE,
  show.centers = TRUE,
  show.colors = TRUE,
  show.grid = TRUE,
  analysis = 1,
  parameter.name = "S",
  options.mcmc = list(),
  restorePar = TRUE,
  mar = NULL,
  angle.3D = 55,
  CI = "ML",
  replicates.CI = 1000,
  show.legend = TRUE,
  ...
)
```

Arguments

`x`	The bone image
`message`	The message to be displayed
`type`	The type of plot; see description
`angle`	Which angle model to show
`show.all.angles`	For periodic type and partial section, should all angles been shown?
`show.centers`	Should the centers be shown?
`show.colors`	Should the background and foreground colors be shown?
`show.grid`	Should the grid be shown?
`analysis`	Name or number of analysis to be plotted
`parameter.name`	The parameter to plot
`options.mcmc`	The option to plot type mcmc output
`restorePar`	If TRUE, restore the par parameter at the exit
`mar`	The margin for type being "model" or "observations"
`angle.3D`	The angle between x and y for 3Dcolors graph
`CI`	Which confidence interval should be plotted: MCMC or ML
`replicates.CI`	How many replicates to estimate CI?
`show.legend`	Should a legend be shown?
`...`	Default parameters for some functions

Details

plot.BoneProfileR displays a bone section

Value

Nothing

Author(s)

Marc Girondot <`marc.girondot@gmail.com`>

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `summary.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
 bone <- BP_OpenImage()
 # or
 path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                              package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
 plot(bone, type="colors")
 bone <- BP_DetectCenters(bone=bone, analysis="logistic")
 plot(bone, type="3Dcolors")
 bone <- BP_EstimateCompactness(bone, analysis="logistic")
 bone <- BP_FitMLCompactness(bone, analysis="logistic")
 plot(bone)
 ############################################
 # Example with comparison between two models
 ############################################
 path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                              package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
 bone <- BP_DetectCenters(bone=bone, analysis="logistic")
 bone <- BP_EstimateCompactness(bone, analysis="logistic")
 bone <- BP_FitMLCompactness(bone, analysis="logistic")
 plot(bone)
 plot(bone, type="observations")
 plot(bone, type="observations+model", analysis=1)
 fittedpar <- BP_GetFittedParameters(bone, analysis="logistic",
                                      ML=TRUE, return.all = FALSE)[, "mean"]
 bone <- BP_DuplicateAnalysis(bone, from="logistic", to="flexit")
 bone <- BP_FitMLCompactness(bone,
                fitted.parameters=c(fittedpar, K1=1, K2=1),
                fixed.parameters=NULL, analysis="flexit")
 compare_AIC(Logistic=BP_GetFittedParameters(bone, analysis="logistic",
                                              ML=TRUE, return.all = TRUE),
             Flexit=BP_GetFittedParameters(bone, analysis="flexit",
                                            ML=TRUE, return.all = TRUE))
 out4p <- plot(bone, type="observations+model", analysis="logistic")
 out6p <- plot(bone, type="observations+model", analysis="flexit")

 ############################################
 # Fit distribution using Bayesian model
 ############################################
 bone <- BP_FitBayesianCompactness(bone, analysis="logistic", n.adapt=100)
 # Test the output - New in version 3.2
 plot(bone, type="mcmc", options.mcmc = list(what="LnL"))
 ##################################################################
 # Clearly the distribution is not stationary; the adaptation was too short
```

```
########################################################################
bone <- BP_FitBayesianCompactness(bone, analysis="logistic", n.adapt=10000)
# Now it is ok
plot(bone, type="mcmc", options.mcmc = list(what="LnL"))
########################################################################
# New in version 3.2
########################################################################
plot(bone, type="mcmc", options.mcmc = list(what="Posterior",
     xlim=c(0.025, 0.035), breaks=seq(from=0.025, to=0.035, by=0.001)),
     parameter.name = "S")
plot(bone, type="mcmc", options.mcmc = list(what="MarkovChain",
                                        ylim=c(0.02, 0.04)),
                                        parameter.name = "S")
########################################################################
# Check the priors and the output
########################################################################
mcmc <- RM_get(x=bone, RMname="logistic", valuename = "mcmc")
priors <- mcmc$parametersMCMC$parameters
parameters <- as.parameters(mcmc, index="median")
########################################################################
# Now it is ok. It can be used
########################################################################
plot(bone, type="observations+model", CI="MCMC")
plot(bone, type="observations+model", CI="ML")
########################################################################

#############################################
# Radial compactness
#############################################
bone <- BP_FitMLRadialCompactness(bone, progressbar=TRUE)
plot(bone, type="radial", parameter.name=c("P", "S"))
plot(bone, type="radial", parameter.name=c("P", "S", "Min", "Max"))
plot(bone, type="radial", parameter.name="observed.compactness")
plot(bone, type="radial", parameter.name="linearized.observed.compactness")

#############################################
# Periodic analysis
# This model can take 10 minutes to be fitted
# And still more if you use large replicates.CI value
#############################################
bone <- BP_FitMLPeriodicCompactness(bone, analysis="logistic", control.optim=list(trace=2),
                                    fitted.parameters=c(par, PSin=0.001, PCos=0.001,
                                    SSin=0.001, SCos=0.001, MinSin=0.001, MinCos=0.001,
                                    MaxSin=0.001, MaxCos=0.001), replicates.CI=2000)
plot(bone, type="periodic", parameter.name="compactness", col=rainbow(128))
plot(bone, type="periodic", parameter.name="compactness")
plot(bone, type="periodic", parameter.name="P", ylim=c(0, 1),
      col=rgb(red = 0.7, green = 0.7, blue = 0.7, alpha = 0.2))
plot(bone, type="periodic", parameter.name="averagemodel")


## End(Not run)
```

`summary.BoneProfileR` *Plot a bone section*

Description

Display information of bone section

Usage

```
## S3 method for class 'BoneProfileR'
summary(
  object,
  analysis = 1,
  periodic.angles = "all",
  periodic.angles.replicate.CI = 2000,
  ...
)
```

Arguments

`object`	The bone image
`analysis`	The analysis to report the compactness
`periodic.angles`	A vector indicating which angle to report for periodic analysis
`periodic.angles.replicate.CI`	Number of replicates to estimate CI
`...`	Not used

Details

summary.BoneProfileR displays a bone section

Value

An invisible list with recorded information

Author(s)

Marc Girondot `<marc.girondot@gmail.com>`

See Also

Other BoneProfileR: `BP_AutoFit()`, `BP_ChooseBackground()`, `BP_ChooseCenter()`, `BP_ChooseForeground()`, `BP_DetectBackground()`, `BP_DetectCenters()`, `BP_DetectForeground()`, `BP_DuplicateAnalysis()`, `BP_EstimateCompactness()`, `BP_FitBayesianCompactness()`, `BP_FitBayesianPeriodicCompactness()`, `BP_FitMLCompactness()`, `BP_FitMLPeriodicCompactness()`, `BP_FitMLRadialCompactness()`, `BP_GetFittedParameters()`, `BP_ListAnalyses()`, `BP_LnLCompactness()`, `BP_OpenImage()`, `BP_Report()`, `Erinaceus_europaeus`, `plot.BoneProfileR()`

Examples

```
## Not run:
# Not run:
library(BoneProfileR)
 bone <- BP_OpenImage()
 # or
 path_Hedgehog <- system.file("extdata", "Erinaceus_europaeus_fem_2-1_small.png",
                              package = "BoneProfileR")
 bone <- BP_OpenImage(file=path_Hedgehog)
 summary(bone)

 bone <- BP_DetectBackground(bone=bone, analysis="logistic")
 bone <- BP_DetectForeground(bone=bone, analysis="logistic")
 bone <- BP_DetectCenters(bone=bone, analysis="logistic")
 bone <- BP_EstimateCompactness(bone, analysis="logistic", cut.angle = 60)
 bone <- BP_FitMLCompactness(bone, analysis="logistic", twosteps=TRUE)
 plot(bone, type="observations+model", analysis="logistic")
 par <- BP_GetFittedParameters(bone, analysis="logistic", ML=TRUE, return.all=FALSE)[, "mean"]
 options(mc.cores=parallel::detectCores())

 ##############################################
 # Periodic analysis
 ##############################################
 bone <- BP_FitMLPeriodicCompactness(bone, analysis="logistic", control.optim=list(trace=2),
                                     fitted.parameters=c(par, PSin=0.001, PCos=0.001,
                                     SSin=0.001, SCos=0.001, MinSin=0.001, MinCos=0.001,
                                     MaxSin=0.001, MaxCos=0.001), replicates.CI=2000)
 bone <- BP_FitBayesianPeriodicCompactness(bone, analysis="logistic", replicates.CI=2000)

 summary(object=bone, analysis="logistic")
 summary(object=bone, analysis="logistic",
         periodic.angles=seq(from=-0.1, to=0.1, length.out=10))

## End(Not run)
```

Index