Package 'measuRing'

May 8, 2026

Type Package

Title Detection and Control of Tree-Ring Widths on Scanned Image Sections

Version 0.5.2

Date 2025-11-03

Maintainer Wilson Lara <wilarhen@gmail.com>

Imports pastecs, png, tiff, dplR

Description Identification of ring borders on scanned image sections from dendrochronological samples. Processing of image reflectances to produce gray matrices and time series of smoothed gray values. Luminance data is plotted on segmented images for users to perform both: visual identification of ring borders or control of automatic detection. Routines to visually include/exclude ring borders on the R graphical devices, or automatically detect ring borders using a linear detection algorithm. This algorithm detects ring borders according to positive/negative extreme values in the smoothed time-series of gray values. Most of the in-package routines can be recursively implemented using the multiDetect() function.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2025-11-03 13:30:02 UTC

Author Wilson Lara [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Contents

`measuRing-package` *Detection and Control of Tree-Ring Widths on Scanned Image Sections*

Description

Identification of ring borders on scanned image sections from dendrochronological samples. Processing of image reflectances to produce gray matrices and time series of smoothed gray values. Luminance data is plotted on segmented images for users to perform both: visual identification of ring borders or control of automatic detection. Routines to visually include/exclude ring borders on the R graphical devices, or automatically detect ring borders using a linear detection algorithm. This algorithm detects ring borders according to positive/negative extreme values in the smoothed time-series of gray values. Most of the in-package routines can be recursively implemented using the multiDetect() function.

Details

The DESCRIPTION file:

Package:	measuRing
Type:	Package
Title:	Detection and Control of Tree-Ring Widths on Scanned Image Sections
Version:	0.5.2
Authors@R:	c(person(given = "Wilson", family = "Lara", role = c("aut", "cre"), email = "wilarhen@gmail.com", com
Date:	2025-11-03
Maintainer:	Wilson Lara <wilarhen@gmail.com>
Imports:	pastecs, png, tiff, dplR
Description:	Identification of ring borders on scanned image sections from dendrochronological samples. Processing
License:	GPL-3
NeedsCompilation:	no
Packaged:	2018-02-20 15:02:58 UTC; wilar
Repository:	CRAN
Date/Publication:	2018-02-20 15:15:55 UTC
Author:	Wilson Lara [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID

Index of help topics:

colNarrow	Narrow rings
crossRings	dplR crossdating
dataSegments	Data segments
grayDarker	Gray extremes
graySmoothed	Smoothed gray
imageTogray	Gray matrix
lagIngray	First-local lag
linearDetect	Linear detection
measuRing-package	Detection and Control of Tree-Ring Widths on Scanned Image Sections
multiDetect	Multiple detection of TRWs
plotSegments	Image segments
reduceList	ring-width object reduction
ringBorders	Ring borders
ringDetect	Single Detection of TRWs
ringSelect	Visual selection
ringWidths	Ring widths

Maintainer: Wilson Lara <wilarhen@gmail.com>

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

`colNarrow` *Narrow rings*

Description

This function can detect narrow rings in a sequence of tree-ring widths in wood (TRWs). This and other in-package functions are recursively implemented by `multiDetect`.

Usage

```
colNarrow(rwidths, marker = 5)
```

Arguments

`rwidths`	a dataframe with the ring widths such as that produced by `ringWidths`.
`marker`	a number from 1 to 10. Those rings with scaled averages greater than or equal to this argument will be identified as narrow rings.

Details

Each ring is averaged with those rings on either side of it (t-1,t,t+1), and averages are divided by the highest computed average in the sample; such quotients are scaled from 10 (the narrowest possible ring) to one (the broadest ring).

Value

character vector with the columns in gray matrix corresponding to the narrow rings (see `ringDetect`, `multiDetect`, and`plotSegments`).

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## (not run) Read one image section in package measuRing:
image1 <- system.file("P105_a.png", package="measuRing")
## (not run) compute a gray matrix from RGB in the image:
gray <- imageTogray(image = image1,ppi=1000)
## (not run) Columns in gray matrix to be included/excluded:
Toinc <- c(196,202,387,1564)
Toexc <- c(21,130,197,207,1444,1484)
## (not run) tree-ring widths:
rwidths <- ringWidths(gray,inclu = Toinc,exclu = Toexc,last.yr=2012)
##(not run) narrow rings:
narrows <- colNarrow(rwidths,marker = 8)
```

`crossRings` *dplR crossdating*

Description

This function implements routines to crossdate TRWs.

Usage

```
crossRings(mdr, smp = 1,
    ncol = 1:length(mdr),
    fun = "corr", ...)
```

Arguments

`mdr`	`list`. Set of detected TRWs such as that produced by `multiDetect`.
`smp`	`numeric` or `character`. Position or name in the set of the sample being crossdated.
`ncol`	`numeric` or `character`. Positions or names in the set of the TRWs used to crossdate the sample.
`fun`	`character`. Function to be implemented. Three functions can be used: `corr.rwl.seg` ('corr'), `ccf.series.rwl` ('ccf'), and `spag.plot` ('spag').
`...`	arguments to be passed to the function.

Value

output of selected function.

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## Paths to three image sections in the package:
img <- system.file(c("P105_a.tif",
                     "P105_b.tif",
                     "P105_d.tif"),
                   package="measuRing")

## Recursive detection:
mrings <- multiDetect(img,
                      last.yr = 2013,
                      auto.det = TRUE,
                      plot = FALSE)

## corr analysis
crossRings(mrings,
           fun = 'corr',
           seg.length = 10,
           bin.floor = 0,
           lag.max = 2,
           make.plot = FALSE)
```

---

dataSegments *Data segments*

Description

Segmented data sets required by function `plotSegments`.

Usage

```
dataSegments(image, segs = 1,
    ...)
```

Arguments

`image`	Either path of an `image` section or an array representing a gray matrix.
`segs`	number of `image` segments.
`...`	arguments to be passed to three functions: `ringWidths`, `ringBorders`, and/or `imageTogray`.

Value

a list with segmented sets of the gray matrix, the ring borders, and the ring widths (see `plotSegments`).

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## (not run) Read one image section in package measuRing:
image1 <- system.file("P105_a.tif", package="measuRing")
## (not run) compute a gray matrix from its RGB:
gray <- imageTogray(image1)
## (not run) Columns in gray matrix to be included/excluded:
Toinc <- c(196,202,387,1564)
Toexc <- c(21,130,197,207,1444,1484)
## (not run) segmented data:
segm <- dataSegments(image1,segs = 3)
lapply(segm,str)
attributes(segm)
```

---

grayDarker *Gray extremes*

Description

This function can detect the extremes of the `smoothed` gray.

Usage

```
grayDarker(smoothed,
    origin = 0, darker = TRUE)
```

Arguments

`smoothed`	a data frame with the `smoothed` gray such as that produced by `graySmoothed`.
`origin`	an `origin` to find the extremes.
`darker`	logical. If TRUE the function finds the negative extremes. If FALSE the positive extremes are detected.

Value

vector with the columns in gray matrix corresponding to the extremes (see `graySmoothed` and `linearDetect`).

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## (not run) read one image section:
image1 <- system.file("P105_a.png", package="measuRing")
## (not run) gray matrix from RGB in image:
gray <- imageTogray(image = image1,ppi = 1000)
## (not run) smoothed gray:
smoothed <- graySmoothed(gray)
## (not run) column numbers of possitive and negative extremes:
posit <- grayDarker(smoothed,darker=FALSE)
nega <- grayDarker(smoothed,darker=TRUE)
str(nega)
```

---

graySmoothed *Smoothed gray*

Description

Averaging, detrending, and smoothing of the columns in a gray matrix.

Usage

```
graySmoothed(image, all = FALSE,
    ...)
```

Arguments

image	character or matrix. Either path of an image section or an array representing a gray matrix.
all	logical. If TRUE the column numbers and moving averages are added to the output.
...	arguments to be passed to imageTogray.

Value

data frame with the smoothed grays. If all is TRUE then the output is extended with the columns in gray matrix, and moving averages.

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## (not run) Read one image section in package measuRing:
image1 <- system.file("P105_a.png", package="measuRing")
## (not run) the smoothed gray:
smoothed <- graySmoothed(image1,ppi=1000)
## (not run) Plot of the smoothed gray:
Smooth <- ts(smoothed)
main. <- 'Smoothed gray'
plot(Smooth,xlab = 'Column', main=main.,
     ylab = 'Smoothed gray',col = 'gray')
```

`imageTogray` *Gray matrix*

Description

This function can compute a gray matrix from the RGB in an `image` section. Such an `image` section can be compressed in either portable network graphics format (png) or tagged `image` file format (tif).

Usage

```
imageTogray(image, ppi = NULL,
    rgb = c(0.3, 0.6,
        0.1), p.row = 1)
```

Arguments

`image`	character. path of an `image` section.
`ppi`	NULL or integer. If NULL the `image` resolution in points per inch is extracted from attributes in `image`. If this attribute is not embedded then users should provide it
`rgb`	vector with three fractions, all of them adding to one, to combine RGB channels into gray matrix.
`p.row`	proportion of rows of gray matrix to be processed.

Value

a gray matrix containing the `image` reflectances.

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## (not run) Read two image sections in package measuRing:
image1 <- system.file("P105_a.tif", package="measuRing")
image2 <- system.file("P105_a.png", package="measuRing")
## (not run) compute a gray matrix:
gray <- imageTogray(image1)
## (not run) - the ppi is embedded in the image:
attributes(gray)
## (not run) but, the ppi is not embedded in image2:
## - imageTogray will return an error:
## (uncoment and run):
## gray2 <- imageTogray(image2)
## attributes(gray2)
## - the ppi should be provided (i.e. ppi = 1200):
gray3 <- imageTogray(image2,ppi = 1200)
attributes(gray3)
##(not run) a plot of the gray matrix
xrange <- range(0:ncol(gray)) + c(-1,1)
yrange <- range(0:nrow(gray)) + c(-1,1)
{plot(xrange,yrange,xlim=xrange,ylim=yrange,xlab='',
      ylab='',type='n',asp=0)
rasterImage(gray,xrange[1],yrange[1],xrange[2],yrange[2])}
```

---

lagIngray *First-local lag*

Description

This function can compute the lag of the first local on the auto-correlation function (`acf`) of smoothed grays.

Usage

```
lagIngray(image, acf = FALSE,
    ...)
```

Arguments

`image`	character or matrix. Either path of an `image` section or an array representing a gray matrix.
`acf`	logical. If TRUE the output is extended with the `acf`.
`...`	arguments to be passed to `imageTogray`.

Value

constant value of the first local on the `acf` of the smoothed gray. If `acf` is TRUE then the computed `acf` is added to the output (see `linearDetect`, and `graySmoothed`).

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## (not run) Read one image sample in folder of package measuRing:
image1 <- system.file("P105_a.tif", package="measuRing")
##(not run) First local in the acf of smoothed grays:
local1 <- lagIngray(image1,acf = TRUE)
##(not run) Plot of first local over the acf:
Flocal <- local1[['local']]
Clocal <- ts(local1[['acf']][Flocal,],start=Flocal)
acf <- ts(local1[['acf']],start=1)
{plot(acf,type='h',col='gray',xlab='Lag',main='First local lag')
points(Clocal,pch=19,cex=0.5)}
```

---

linearDetect *Linear detection*

Description

Function for developing linear detection of ring borders.

Usage

```
linearDetect(smoothed,
    origin = 0, darker = TRUE)
```

Arguments

smoothed	a data frame with smoothed grays such as that produced by graySmoothed.
origin	numeric. an origin in smoothed gray to find the ring borders.
darker	logical. If TRUE the algorithm uses the negative extremes on smoothed grays to detect the ring borders. If FALSE the possitive extremes are used.

Value

vector with column numbers in gray matrix of the detected ring borders (see grayDarker, and graySmoothed).

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## (not run) Read one image section in package measuRing:
image1 <- system.file("P105_a.tif", package="measuRing")
## (not run) smoothed gray:
smoothed <- graySmoothed(image1)
## linear detection:
borders <- linearDetect(smoothed)
str(borders)
```

`multiDetect` *Multiple detection of TRWs*

Description

This function recursively detects TRWs in sets of scanned images of wood or gray matrices.

Usage

```
multiDetect(image, ...)
```

Arguments

`image`	`character` or `list`. Vector of Paths to the `image` files or set of gray matrices.
`...`	Further arguments to be passed to `ringDetect` and `ringSelect`.

Value

`list`. Set of `ringDetect` calls.

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## Paths to three image sections in the package:
img <- system.file(c("P105_a.tif",
                     "P105_b.tif",
                     "P105_d.tif"),
                   package="measuRing")

## Recursive detection. Arbitrary ring borders and different years
## of formation of last rings in the images years are specified:
mrings <- multiDetect(img,
                      inclu = list(c(1:40),c(1:30),c(1:41)),
                      last.yr = list(2014, 2013, 2012),
```

```
                    auto.det = c(FALSE,TRUE,FALSE),
                    plot = FALSE)
str(mrings)

## Updating the call in mrings using new arguments:
mrings1 <- update(mrings,
                  exclu = list(c(1:4),c(1:4),c(1:4)),
                  last.yr = 2016)
```

---

plotSegments *Image segments*

Description

One or several plots of consecutive segments of gray matrix and smoothed grays.

Usage

```
plotSegments(image, ratio = NULL,
    marker = NULL, col.marker = "red",
    tit = TRUE, plot = TRUE,
    ...)
```

Arguments

image	character or matrix. Either path of an image section or an array representing a gray matrix.
ratio	NULL or vector with two values representing the aspect of the plots (height, and width). If NULL the default aspect in par() is used.
marker	NULL or a number from 1 to 10 as explained in colNarrow. If numeric then three kind of markers are indicated: those narrow rings with averages major than marker, chronological markers (decades, centuries, and millenia), and the column numbers in gray matrix of the ring borders. The markers are highlighted with the color in col.marker. If NULL no markers are highlighted.
col.marker	color of the markers.
tit	logical or character. A title for the plots. If TRUE the main title is the image name. For more than 1 segment the main title ends with the segment number.
plot	logical. If TRUE the image segments are plotted.
...	arguments to be passed to four functions: dataSegments, ringWidths, ringBorders, and/or imageTogray.

Value

the image segments and a list such as that produced by dataSegments.

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## (not run) Read one image sample in folder of package measuRing:
image1 <- system.file("P105_a.tif", package="measuRing")
## column numbers to be included/avoided:
Toinc <- c(196,202,387,1564)
Toexc <- c(21,130,197,207,1444,1484)
##(not run) Plotting of five image segments:
plots <- plotSegments(image1,rgb=c(0.5,0,0.5),last.yr=2011,
    marker=8,segs=3,inclu = Toinc,exclu = Toexc)
## plots <- plotSegments(rwidths,segs = 4,marker=8)
## (not run) kill all the image segments:
graphics.off()
```

---

reduceList *ring-width object reduction*

Description

ring-width objects are reduced to dplR chronologies.

Usage

```
reduceList(mls, name.ls = "ringWidths",
    empty.rm = TRUE)
```

Arguments

mls	List. Object from multiDetect
name.ls	Character. name of the list to be reduced.
empty.rm	Logical. Remove empty lists.

Value

data frame in wide format with the ring widths.

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## Paths to three image sections in the package:
img <- system.file(c("P105_a.tif",
                     "P105_b.tif",
                     "P105_d.tif"),
                   package="measuRing")

## Recursive detection (arbitrary ring borders and formation years
## are included):
mrings <- multiDetect(img,
                      inclu = list(c(1:40),c(1:30),c(1:41)),
                      last.yr = list(2014, 2013, 2012),
                      auto.det = c(FALSE,TRUE,FALSE),
                      plot = FALSE)
## Reducing the processed ring withs
wide <- reduceList(mrings)
tail(wide)
```

`ringBorders` *Ring borders*

Description

This function can find the ring borders in a gray matrix.

Usage

```
ringBorders(image, auto.det = TRUE,
    darker = TRUE, origin = 0,
    inclu = NULL, exclu = NULL,
    ...)
```

Arguments

`image`	character or matrix. Either path of an `image` section or an array ##representing a gray matrix.
`auto.det`	logical. If TRUE the linear detection is implemented (see `linearDetect`).
`darker`	logical. If TRUE the algorithm uses the negative extremes on smoothed grays to detect the ring borders. If FALSE the possitive extremes are used.
`origin`	numeric. an `origin` in smoothed gray to find the ring borders.
`inclu`	NULL or vector with column numbers in gray matrix, other than those automatically detected, to be considered as ring borders.If NULL no column numbers are included.
`exclu`	NULL or vector with column numbers in gray
`...`	arguments to be passed to `imageTogray`.

Value

a data frame with the smoothed grays and the identified ring borders (see grayDarker, graySmoothed, and linearDetect).

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## (not run) Read one image sample in folder of package
## measuRing:
image1 <- system.file("P105_a.tif", package="measuRing")
## column numbers in gray matrix to be included/avoided:
Toinc <- c(196,202,387,1564)
Toexc <- c(21,130,197,207,1444,1484)
##(not run) the ring borders:
borders <- ringBorders(image1,inclu = Toinc,exclu = Toexc)
str(borders)
##(not run) Plot of smoothed grays with the ring borders:
Smooth <- ts(borders[,1])
inclupix <- subset(borders,borders%in%TRUE)
inclucol <- as.numeric(rownames(inclupix))
xyborders <- data.frame(column=inclucol,smooth=inclupix[,1])
y.lim <- c(-0.05,0.05)
main. <- 'Ring borders'
{plot(Smooth,xlab = 'Column',ylab = 'Smoothed gray',
      main=main.,col = 'darkgoldenrod1')
 points(xyborders[,1],xyborders[,2],pch=19,cex=0.5,col='orangered')}
```

---

ringDetect *Single Detection of TRWs*

Description

This function assists in the detection of TRW (mm) in a scanned image (.tif or .png) or gray matrix, evaluating other required functions and plotting the outputs in graphics devices. The function can be combined with ringSelect to visually detected TRWs. Nevertheless, the complete measurement procedure of TRW with the package can be performed by multiDetect.

Usage

```
ringDetect(image, ...)
```

Arguments

image	character or matrix. Vector of path to the image section or a gray matrix such as that produced by imageTogray.
...	arguments to be passed to other functions, see section of Details.

Details

The scanned sample should correspond to a horizontal window of wood with the bark side located towards the left area of the image, and the pit side towards the right. The image section may not necessarily contain both bark and pit, see images of the Examples. The image should contain Red, Gren, and Blue channels (rgb) and be compressed in any of two file formats: tif or png. These are easily obtained by scanning wood samples with a conventional scanner and extracting a horizontal image section. Five functions are internally implemented: plotSegments, dataSegments, ringWidths, ringBorders, and/orimageTogray. These are controlled using the following arguments:

- ppi = NULL: image resolution in points per inch. If NULL the ppi is extracted from the image attributes. If the ppi is not embedded in the image, then thos argument should be provided;
- rgb = c(0.3,0.6,0.1): vector of three fractions, all of them adding to one, to combine the rgb into a gray matrix. Defaults correspond to the rgb-standard in the luminosity function (Russ, 2006): green light contributes the most to the intensity perceived by humans, and blue light the least;
- p.row = 1: Proportion of rows in the central portion of the gray matrix to be processed;
- last.yr = NULL: NULL or integer. Year of formation of the newest ring. If NULL then the rings are numbered from one (right) to the number of detected rings (left);
- auto.det = TRUE: logical. If TRUE then an algorithm for automatic detection is implemented, see linearDetect;
- darker = TRUE: logical. If TRUE then the algorithm uses the negative extremes on smoothed grays to detect the ring borders. If FALSE the positive extremes are used instead;
- origin = 0: An origin along central portion of the smoothed gray to find the ring borders. This value could help to avoid noisy areas during the visual detection process;
- inclu = NULL: NULL or vector with column numbers in the gray matrix, other than those automatically detected, to be considered as ring borders. If NULL no column numbers are included;
- exclu = NULL: NULL or vector with column numbers in gray matrix of those ring borders to be excluded from the analysis. If NULL, no ring borders are excluded;
- plot = TRUE: logical. If TRUE then a plot is produced;
- segs = 1: Number of image segments to be plotted;
- ratio = NULL: NULL or numeric vector of two values representing the aspect ratio of the plots (height, and width). If NULL default in par is used;
- marker = NULL: NULL or integer vector with any value from 1 to 10. The rings are averaged with those rings on either side of it and the averages are scaled from ten (the narrowest possible ring) to one (the broadest ring). The narrow rings with averages larger than marker as well as other chronological markers (decades, centuries, and millennia), are highlighted with red pinpricks;

- col.marker = 'red': color of the markers;
- tit = TRUE: logical or character. A title for the plots. If TRUE the main title is the image name. For more than 1 segment the main title ends with the segment number. This argument does not work in multiDetect.

If users run R from Interactive Development Environments (IDE) aiming to segment the image section (segs > 1), they should be sure that such environments support multiple graphics devices. If the argument image is a gray matrix, then other arguments passed to imageTogray will be ignored. The function can be combined with ringSelect to visually include/exclude ring borders in the plot output, see examples in the ringSelect function. See multiDetect for recursive implementation of this function.

references«

- Lara W., F. Bravo, and S. Carlos. 2015. measuRing: An R package to measure tree-ring widths from scanned images. Dendrochronologia, 34: 43-50;
- Russ, J.C., 2006. The Image Processing Handbook, Fifth Edition. CRC Press, Boca Raton, 817 pp.

Value

list of data frames with ring widths and ring borders such as these produced by ringWidths, and ringBorders.

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
image1 <- system.file("P105_a.tif", package="measuRing")
## (not run) Initial diagnostic:
detect1 <- ringDetect(image1,segs=3)
## (not run) Updating ringDetect to chage arguments;
## and flagged rings
detect1 <- update(detect1,marker=8)
## (not run) Some noise in smoothed gray can be avoided
## by moving the origin:
detect1 <- update(detect1,origin = -0.03)
## (not run) columns 21 and 130 are not considered now.
##
## (not run) Choose other columns in gray matrix (see ringSelect);
## (not run) graphical devices from ringDetect should be active!
## (not run) Including columns:
## (uncomment and run):
## detect1 <- update(detect1)
## Toinc <- ringSelect(detect1)
## detect1 <- update(detect1, inclu = Toinc)
## or, include the next columns:
Toinc <- c(202,387,1564)
```

```
detect1 <- update(detect1,inclu = Toinc)
## (not run) Object detec1 is updated with Toinc;
##
## (not run)  ring borders to be excluded:
## (uncomment and run):
## detect1 <- update(detect1)
## Toexc <- ringSelect(detect1,any.col = FALSE)
## detect1 <- update(detect1,exclu=Toexc)
## or, exclude the nex columns:
Toexc <- c(208,1444,1484)
detect1 <- update(detect1,exclu = Toexc)
##
## (not run) Final arguments:
detect2 <- update(detect1,last.yr=2011,marker = 8)
str(detect2)
##
## (not run) kill previous plot:
graphics.off()
##
## (not run) Tree-ring widths and attributes:
rings <- detect2$'ringWidths'
##
## (not run) Plot of the tree-ring witdths:
maint <- 'Hello ring widths!'
plot(rings,ylab = 'width (mm)',type='l',col = 'red',main=maint)
```

`ringSelect` *Visual selection*

Description

This function can include and exclude ring borders in plot outputs from `ringDetect` or `plotSegments`. The function is mapped by `multiDetect` for recursive processing of image sections.

Usage

```
ringSelect(rdetect, any.col = TRUE)
```

Arguments

`rdetect`	a list containing data frames of ring widths and ring borders such as that produced by `ringDetect`.
`any.col`	logical. If FALSE only those column numbers in gray matrix previouly identified as ring borders can be selected.

Details

Columns in gray matrix are either identified and stored by left-clicking the mouse over the central axis of a gray image in the plot output; pixel numbers of just added ring borders are highlighted on the gray raster. The raphics devices are sequentially closed by right-clicking the mouse. After a graphics device has been closed, the graphics device of the following segment is activated, and visual selection on such a new segment can be performed. Closing the graphics devices with other procedures will stop the selection of ring borders. This detection process can be recursively developed on several image section using `multiDetect`.

Value

vector with column numbers in gray matrix of the identified ring borders.

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Lara W., F. Bravo, and S. Carlos. 2015. measuRing: An R package to measure tree-ring widths from scanned images Dendrochronologia, 34: 43-50.

Examples

```
## Read one image in package folder:
image1 <- system.file("P105_a.tif", package="measuRing")
## (not run) Initial diagnostic:
detect1 <- ringDetect(image1,segs=2,marker=7)
##
## (not run) Choose other columns in gray matrix (see ringSelect);
## (not run) graphical devices from ringDetect should be active!
## (not run) Including columns:
##
## (uncomment and run):
## Toinc <- ringSelect(detect1)
## detect1 <- update(detect1, inclu = Toinc)
##
## (not run)  ring borders to be excluded:
## (uncomment and run):
## Toexc <- ringSelect(detect1,any.col = FALSE)
## detect1 <- update(detect1, exclu=Toexc)
## (not run) kill previous plot:
graphics.off()
```

`ringWidths` *Ring widths*

Description

This function can compute the ring widths (mm) from the ring borders detected on an `image` section.

Usage

```
ringWidths(image, last.yr = NULL,
    ...)
```

Arguments

`image`	character or matrix. Either path of an `image` section or an array representing a gray matrix.
`last.yr`	year of formation of the newest ring. If NULL then the rings are numbered from one (right) to the number of detected rings (left).
`...`	arguments to be passed to two functions: `ringBorders`, and/or `imageTogray`.

Value

data frame with the ring widths.

Author(s)

Wilson Lara <wilarhen@gmail.com> [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Carlos Sierra [aut] (ORCID: <https://orcid.org/0000-0003-0009-4169>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## (not run) Read one image section:
image1 <- system.file("P105_a.tif", package="measuRing")
## (not run) columns in gray matrix to be included/excluded:
Toinc <- c(196,202,387,1564)
Toexc <- c(21,130,197,207,1444,1484)
## (not run) tree-ring widths
rwidths <- ringWidths(image1,inclu = Toinc,exclu = Toexc,last.yr=NULL)
str(rwidths)
##plot of computed tree-ring widths:
maint <- 'Hello ring widths!'
plot(rwidths,type='l',col = 'red',main = maint,
     xlab = 'Year',ylab = 'Width (mm)')
```

Index