Package 'measuRing'

March 23, 2015

Type Package

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

Version 0.3

Author Wilson Lara, Carlos Sierra, Felipe Bravo

Date 2015-03-23

Maintainer Wilson Lara <wilarhen@gmail.com>

Depends pastecs, png, tiff

Description Identification of ring borders on scanned image sections from dendrochronological samples. Processesing of image reflectances to produce gray matrices and a 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 device, 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.

License GPL-3

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2015-03-23 10:49:59

R topics documented:

`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. Processesing of image reflectances to produce gray matrices and a 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 device, 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.

Details

Package:	measuRing
Type:	Package
Title:	Detection and Control of Tree-Ring Widths on Scanned Image Sections
Version:	0.3
Author:	Wilson Lara, Carlos Sierra, Felipe Bravo
Date:	2015-03-23
Maintainer:	Wilson Lara <wilarhen@gmail.com>
Depends:	pastecs, png, tiff
License:	GPL-3
LazyData:	TRUE

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

`colNarrow` *Narrow rings*

Description

This function can detect narrow rings in an object such as that produced by ringWidths.

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, andplotSegments).

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

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)
```

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 ringWidths.

Value

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

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

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 possitive extremes are detected.

Value

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

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

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 argument all is TRUE the output is extended the with the columns in gray matrix, and the moving averages.

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

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 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, Carlos Sierra, Felipe Bravo

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, Carlos Sierra, Felipe Bravo

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, Carlos Sierra, Felipe Bravo

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*

Description

This function provides tools to recursively detect the ring borders in multiple image sections.

Usage

```
multiDetect(pattern, is.png = FALSE, from = 1, to = "all", inclu.dat = NULL,
    ...)
```

Arguments

`pattern`	character with a common pattern in the names of the image sections.
`is.png`	logical. If FALSE the tif images in working directory are processed.
`from`	character with a complementary pattern, or position number in folder, of the initial image to be processed.
`to`	character with a complementary pattern, or position number in folder, of the final image to be processed. If this argument is 'all' then all the images matching the argument in pattern are processed.
`inclu.dat`	a data frame such as that contained in the ouput of this same function with the column numbers to be updated.
`...`	arguments to be passed to `ringDetect` (ppi, last.yr, rgb, p.row, auto.det, darker, origin, inclu, exclu, and plot), or to `plotSegments`(segs, marker, col.marker, ratio, and tit).

Details

Users running R from IDEs and aiming to develop visual control on several image segments should be sure that such environments support multiple graphics devices (see `ringDetect`).

Value

list with three data frames: the tree-ring widths, the column numbers of the detected ring borders, and the narrow rings (see `ringBorders`, `ringDetect`, and `plotSegments`).

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

Examples

```
## (not run) Set working directory:
setwd(system.file(package="measuRing"))
## (not run) List the tif images the folder:
list.files(path=getwd(),pattern='.tif')

## (not run) run multiDetect:
## -provide at least one argument of ringDetect
tmp <- multiDetect(pattern = 'P105',last.yr=2012,plot = FALSE)
##
## (not run) Excluding/changing some column numbers in tmp:
dd <- tmp$colNames
ddtest <- dd #to be compared with final outputs
dd[dd$year%in%1999:2012,] <- NA
tail(dd,20)
tmp1 <- update(tmp,inclu=dd,auto.det=FALSE)
dm <- tmp1$colNames
dmtest <- dm #to be compared with final outputs
##
## (not run)changing columns from tmp with visual control
## -choose five or six rings at the bark side and later
## exclude any one of them:
tmp2 <- update(tmp,plot=TRUE,to='_a',segs = 1,auto.det=FALSE)
dm2 <- tmp2$colNames
newm <- merge(dm,dm2,by='year',all.x=TRUE)
dm[,'P105_a'] <- newm[,ncol(newm)]
tmp3 <- update(tmp,inclu=dm,plot=FALSE,auto.det=FALSE)
dm3 <- tmp3$colNames
## compare initial and final columns in gray matrix
tail(ddtest,15)
tail(dmtest,15)
tail(dm3,15)
```

`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 dataSegments.

Value

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

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

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()
```

---

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, Carlos Sierra, Felipe Bravo

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*

Description

This function can find the tree-ring widths in an scanned image section by visually including/excluding ring borders, or automatically detecting the rings using a linear detection algorithm.

Usage

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

Arguments

`image`	character or matrix. Either path of an image section or an array ##representing a gray matrix.
`...`	arguments to be passed to `plotSegments`.

Details

If users run R from Interactive Development Environments (IDE), they should be sure that such environments support multiple graphics devices. If argument `image` is a gray matrix, then other arguments passed to `imageTogray` will be ignored.

Value

borders such as these produced by `ringWidths`, and `ringBorders`).

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

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 is used to visually include/exclude ring borders; consequently, graphics devices from `ringDetect` or `plotSegments` should be active.

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 gray image; pixel numbers of just added ring borders are highlighted on such a 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.

Value

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

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

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 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 ringBorders.

Value

data frame with the ring widths.

Author(s)

Wilson Lara, Carlos Sierra, Felipe Bravo

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