Package 'xRing'

October 14, 2022

Type Package

Title Identify and Measure Tree Rings on X-Ray Micro-Density Profiles

Version 0.1.1

Description Contains functions to identify tree-ring borders based on X-ray micro-density profiles and a Graphical User Interface (GUI) to visualize density profiles and correct tree-ring borders. Campelo F, Mayer K, Grabner M. (2019) <doi:10.1016/j.dendro.2018.11.002>.

Depends R (>= 3.5)

Suggests detrendeR

Imports dplR, imager, tcltk2, tkRplotR

License GPL (>= 2)

ByteCompile yes

RoxygenNote 7.1.2

NeedsCompilation no

Author Filipe Campelo [aut, cre, cph]
(<https://orcid.org/0000-0001-6022-9948>),
Konrad Mayer [ctb]

Maintainer Filipe Campelo <fcampelo@ci.uc.pt>

Repository CRAN

Date/Publication 2022-04-22 12:20:02 UTC

R topics documented:

`addRing` *Add Tree-Ring Border(s)*

Description

Add a tree-ring border by defining the position of the new border

Usage

```
addRing(object, x, series = NULL)
```

Arguments

`object`	an object of class "xRingList" or "xRing"
`x`	the position (number of the resp. pixel(s)) to set the new tree-ring border
`series`	the name of the series to be changed when the `object` is "xRingList", by default is `NULL`

Value

a "xRing" or "xRingList" object with a tree-ring border added at the position `x` for the series given by `series` argument

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi <- detectRings(PaPiRaw, PaPiSpan)
plot(PaPi$"AFO1001a")
PaPi$AFO1001a <- removeRing(PaPi$AFO1001a, 47)
plot(PaPi$"AFO1001a")
PaPi <- addRing(PaPi, series = "AFO1001a", x = 47)
plot(PaPi$"AFO1001a")
```

`calibrateFilm` *Calibrate Film*

Description

Convenience function to do the whole calibration of a densitometry image in one function call internally calling getSteps and fitCalibrationModel

Usage

```
calibrateFilm(
  im,
  thickness = stepIncrease(0.24, 7),
  density = 1.2922,
  plot = TRUE,
  auto = FALSE,
  nPixel = 50,
  plotAuto = FALSE,
  ...
)
```

Arguments

`im`	a grayscale image
`thickness`	a vector specifying the thickness of the calibration wedge at each step
`density`	the density of the reference material (i.e. the calibration wedge)
`plot`	if TRUE the calibration model is displayed
`auto`	logical. If TRUE, automatic detection of the steps given a line is carried out. Use with care
`nPixel`	if 'auto = TRUE': number of pixels gives the line width
`plotAuto`	if TRUE the automatic detection of the grayscale values is displayed
`...`	further arguments to be passed to loess

Value

an object of class ‘loess‘ representing the film calibration

See Also

getSteps

Examples

```
if (interactive()) {
  # read a sample file
  im <- imRead(file = system.file("img", "AFO1046.1200dpi.png", package = "xRing"))

  # display the image
  imDisplay(im)

  # calibrate the film:
  calibration <- calibrateFilm(im,
    thickness = stepIncrease(0.24, 7),
    density = 1.2922,
    plot = TRUE
  )
}
```

---

combineFrag *Combine Fragments*

Description

This function combines fragments by series

Usage

```
combineFrag(x, frag = NULL)
```

Arguments

x	an "xRingList" object
frag	integer, defines the character position within the series name that identifies fragments. If NULL the function considers series with names having one more character as fragments

Value

an object of class "xRingList" with merged fragments

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi <- detectRings(PaPiRaw, PaPiSpan)
PaPi.merge <- combineFrag(PaPi, frag = 9)
```

`correctRings` *Correct Tree-Ring Borders Interactively*

Description

A Graphical User Interface (GUI) to correct tree-ring borders

Usage

```
correctRings(x, chrono = NULL)
```

Arguments

`x`	an `xRingList` object
`chrono`	a data.frame with a reference chronology, if `NULL` a reference chronology is calculated using tree-ring width series from `x`

Details

This function uses the `tkRplot` function (tkRplotR package) to interact with X-ray microdensity profiles.

Value

an `xRingList` object

Examples

```
if (interactive()) {
  data(PaPiRaw)
  data(PaPiSpan)
  PaPi <- detectRings(PaPiRaw, PaPiSpan)
  PaPiCorrect <- correctRings(PaPi)
}
```

detectEwLw *Detect the Transition from Earlywood to Latewood*

Description

This function detects the end of earlywood and the start of latewood

Usage

```
detectEwLw(x, ew = 0.5, lw = NULL)
```

Arguments

x	an "xRingList" object
ew	defines the end of earlywood as the ratio of the density range. The default value is 0.5, which means that the end of earlywood is placed at the point where the density is half the range between the minimum and maximum density values within an annual ring
lw	defines the start of latewood, the default value is NULL. When ew is 0.5 and lw is NULL the boundary between earlywood and latewood is placed where the density is half the range between the minimum and maximum density values within an annual ring

Value

an "xRingList" object with limits.ew and limits.lw added.

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi <- detectRings(PaPiRaw, PaPiSpan)
PaPi.merge <- combineFrag(PaPi, frag = 9)
PaPiRings <- detectEwLw(PaPi.merge, ew = 0.5)
```

detectRings *Detect Tree-Ring Borders*

Description

This function identifies tree-ring borders on X-ray microdensity profiles.

Usage

```
detectRings(x, y = NULL, k = 3, minTrw = 3, threshold = 0.215)
```

Arguments

x	a dataframe with X-ray microdensity profiles or an "xRingList" object
y	a dataframe with the first and last year in columns and the series in rows, is NULL by default
k	width of the rolling window to find the local maximum and minimum (for more details please see the help of getBorders function)
minTrw	integer width of the narrowest tree-ring, rings narrower than this value will not be considered
threshold	the minimum difference between local maximum and minimum density to identify a tree-ring border

Details

This function uses the getBorders function to identify tree-ring borders based on the difference between local maximum and minimum density.

Value

detectRings returns an "xRingList" object, an S3 class with "xRing" lists as members, with the following elements:

span first and last year

trw gives the tree-ring width

name a string giving the series name

limits a vector with the position of the tree-ring borders

years a vector with the calendar year

profile.raw a vector with the input

See Also

getBorders

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi <- toxRingList(PaPiRaw, PaPiSpan)
PaPi <- detectRings(PaPi)
# give the same
PaPi <- detectRings(PaPiRaw, PaPiSpan)
# Because the last year is not supplied the last year for all series is the last calendar year
# as.numeric(format(Sys.time(), "%Y"))-1
PaPi <- detectRings(PaPiRaw)
```

`fitCalibrationModel` *Fit a Calibration Curve*

Description

Fit a model to calibrate a film from X-ray densitometry.

Usage

```
fitCalibrationModel(
  grayvalues,
  thickness = stepIncrease(0.24, 7),
  density = 1.2922,
  plot = TRUE,
  ...
)
```

Arguments

`grayvalues`	a numeric vector containing the gray values of the steps of the calibration wedge at various thicknesses given by the argument 'thickness'
`thickness`	a vector specifying the thickness of the calibration wedge at each step.
`density`	the density of the reference material
`plot`	if TRUE the calibration model is displayed
`...`	further arguments to be passed to loess

Value

an object of class 'loess' representing the film calibration

See Also

getSteps

Examples

```
if (interactive()) {
  # read a sample file
  im <- imRead(file = system.file("img", "AFO1046.1200dpi.png", package = "xRing"))

  # display the image
  imDisplay(im)

  # get the grayvalues from the calibration wedge on the film
  grayvalues <- getSteps(im, 7)

  # calibrate the film by fitting a model:
```

```
  calibration <- fitCalibrationModel(grayvalues,
    thickness = stepIncrease(0.24, 7),
    density = 1.2922,
    plot = TRUE
  )
}
```

---

getBorders *Get Tree-Ring Borders*

Description

Identify tree-ring borders

Usage

```
getBorders(x, k = 3, minTrw = 3, threshold = 0.215, addLastBorder = FALSE)
```

Arguments

x	an object of class "xRing"
k	integer; width of the rolling window
minTrw	integer; width of the narrowest tree-ring, rings narrower than this value will not be considered
threshold	the minimum difference between the local maximum and minimum density to detect tree-ring borders
addLastBorder	logical; if FALSE the last border is not added. If TRUE the last border is placed at the position of the last value.

Details

This function uses local maximum and minimum densities in order to detect tree-ring borders.

Value

The getBorders function returns an object of lass "xRing" including the following elements:

names a string giving the series name

span the first and last year

trw a data.frame with tree-ring width

limits a vector with the position of the tree-ring borders

years a vector with the calendar year

profile.raw a vector with the raw X-ray values

profile a vector with the the smoothed X-ray values (if is supplied in the input)

Examples

```
data("PaPiRaw")
data("PaPiSpan")
AFO1001a <- toxRing(PaPiRaw, PaPiSpan, "AFO1001a")
AFO1001a <- getBorders(AFO1001a)

AFO1001a <- toxRing(PaPiRaw, seriesName = "AFO1001a")
AFO1001a <- getBorders(AFO1001a)
```

`getDensity` *Get Density Values*

Description

Get wood density parameters by tree-ring.

Usage

```
getDensity(x)
```

Arguments

x	a "xRingList" or "xRing" object

Value

a "xRingList" or "xRing" object with density values c("Dmean", "Dmin", "Dmax", "Dew", "Dlw") for each ring

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi <- detectRings(PaPiRaw, PaPiSpan)
PaPi.merge <- combineFrag(PaPi, frag = 9)
PaPiRings <- detectEwLw(PaPi.merge, ew = 0.5)

PaPi <- detectRings(PaPiRaw, PaPiSpan)
PaPiRings <- detectEwLw(PaPi, ew = 0.5)

# xRingList object
PaPiDen <- getDensity(PaPiRings)

PaPiDen$AFO1001a[]
PaPiDen$AFO1001a$density
```

```
# xRing object
PaPi_AFO1001a <- getDensity(PaPi$AFO1001a)
# the same
PaPi_1 <- getDensity(PaPi[[1]])
identical(PaPi_AFO1001a, PaPi_1)

# do not work for PaPi[1]
# class(PaPi[1])
# getDensity(PaPi[1]) # 'list' class
```

`getRwls` *Get Data-Frames With Ring Width and Density Values*

Description

Produce a list with 8 data.frames (trw, ew, lw, Dmean, Dew, Dlw, Dmin, Dmax) that can be used by other packages (dplR, detrendeR)

Usage

```
getRwls(x)
```

Arguments

`x`	an "xRingList" object

Value

a list with 8 elements:

trw a data.frame with tree-ring widths

ew a data.frame with earlywood widths

lw a data.frame with latewood widths

Dmean a data.frame with mean tree-ring density

Dew a data.frame with mean earlywood density

Dlw a data.frame with mean latewood density

Dmin a data.frame with the minimum ring density

Dmax a data.frame with the maximum ring density

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi <- detectRings(PaPiRaw, PaPiSpan)
PaPi <- combineFrag(PaPi)
PaPi <- detectEwLw(PaPi)
rwls <- getRwls(PaPi)
names(rwls)
library(dplR)
rwl.report(rwls$trw)
library(detrendeR)
RwlInfo(rwls$trw)
```

---

getSteps *Select the Steps of a Calibration Wedge Interactively*

Description

Obtain the Grayvalue of Each Step of a Calibration Wedge

Usage

```
getSteps(im, nSteps = NULL, auto = FALSE, nPixel = 50)
```

Arguments

im	an image.
nSteps	number of steps of the calibration wedge to obtain grayvalues from.
auto	logical. If TRUE, automatic detection of the steps given a line is carried out. Use with care.
nPixel	gives the line width when 'auto = TRUE'

Value

a numeric vector

Examples

```
if (interactive()) {
  # read a sample file
  im <- imRead(file = system.file("img", "AFO1046.1200dpi.png", package = "xRing"))

  # display the image
  imDisplay(im)

  # get the grayvalues from the calibration wedge on the film
```

```
  steps <- grayvalues <- getSteps(im, 7) # select 7 ROIs
  steps1 <- grayvalues <- getSteps(im, 7, auto = TRUE) # select a single ROI
  cor(steps, steps1)
}
```

---

imCrop *Crop Image Interactively*

Description

A GUI for cropping an image

Usage

```
imCrop(im)
```

Arguments

im	a cimg object

Value

a cropped image

Examples

```
if (interactive()) {
  file_path <-
    system.file("img", "AFO1046.1200dpi.png", package = "xRing")
  im <- imRead(file_path)
  print(dim(im))
  im_crop <- imCrop(im)
  print(dim(im_crop))
}
```

imDisplay *Display Image Using tcltk Package*

Description

xRing

Usage

```
imDisplay(im, zoom = NULL, title = NULL)
```

Arguments

im	an image (an object of class "cimg")
zoom	the zoom factor (ratio), for zoom = 1 the image is shown with no zoom (original size), when zoom is less than 1 the image is zoomed out. The default value of zoom is NULL.
title	the window title

Value

a tcltk object

Examples

```
if (interactive()) {
  file_path <- system.file("img", "AFO1046.1200dpi.png", package = "xRing")
  im <- imRead(file_path)
  tkWin <- imDisplay(im, zoom = .25)
  tkWin$env$ZOOM # 4 means 25% zoom
}
```

imRead *Load Image From a File*

Description

Load an image using the load.image function from imager package

Usage

```
imRead(file)
```

Arguments

file	path to file

Value

an object of class "cimg"

See Also

load.image

Examples

```
if (interactive()) {
  file_path <- system.file("img", "AFO1046.1200dpi.png", package = "xRing")
  im <- imRead(file_path)
  imDisplay(im)
}
```

---

measureProfiles *Measure Profiles Interactively*

Description

Several profiles can be selected in an image and a calibration for that image is used to convert pixels into wood density

Usage

```
measureProfiles(im, nPixel = 50, cal = NULL)
```

Arguments

im	an image
nPixel	the line width
cal	calibration

Value

an xRingList object with all xRing objects

Examples

```
if (interactive()) {
  # read a sample file
  im <- imRead(file = system.file("img", "AFO1046.1200dpi.png", package = "xRing"))

  # to display the image
  imDisplay(im)

 cal1 <- calibrateFilm(im, thickness = stepIncrease(0.24, 7), density = 1.2922, plot = TRUE)
```

```
  profiles <- measureProfiles(im, cal = cal1)
}
```

PaPiRaw *PaPiRaw*

Description

A dataframe with 44 series of wood density (g/m^3).

Usage

```
data("PaPiRaw")
```

Format

A data.frame containing 44 series in columns and 3111 values of wood density in rows.

Examples

```
data(PaPiRaw)
plot(na.omit(PaPiRaw[,1]), type="l", ann = FALSE)
```

PaPiSpan *PaPiSpan*

Description

A dataframe giving the first and the last year of 44 series. The row names give the name of series.

Usage

```
data("PaPiSpan")
```

Format

A data frame with 44 observations on the following 2 variables.

`first` a numeric vector giving the first year

`last` a numeric vector giving the last year

Examples

```
data(PaPiSpan)
head(PaPiSpan)
```

plot *Plot xRing and xRingList Objects*

Description

Plot method for objects of class "xRing" and "xRingList".

Usage

```
## S3 method for class 'xRing'
plot(x, years = NULL, EwLw = TRUE,  xlim = NULL, ylim = NULL, ...)

## S3 method for class 'xRingList'
plot(x, series = 1, years = NULL, EwLw = TRUE,  xlim = NULL, ylim = NULL, ...)
```

Arguments

x	an object of class "xRing" or "xRingList".
years	the years to be plotted, if NULL the whole time span is plotted.
EwLw	logical. If TRUE the earlywood and latewood boundaries and width is plotted.
xlim	vector of length 2 giving the x limits for the plot.
ylim	the y limits of the plot.
...	other arguments to be passed to plotRings function
series	gives the name (or the index) of the series to be plotted, by default is 1 (i.e., the first series)

Value

None.

See Also

plotRings

Examples

```
data(PaPiRaw)
data(PaPiSpan)

PaPi <- detectRings(PaPiRaw, PaPiSpan)
class(PaPi)

PaPiRings <- detectEwLw(PaPi, ew = 0.5)
plot(PaPiRings, series = "AFO1001a")

PaPiRings1 <- detectEwLw(PaPi, ew = 0.35, lw = 0.55)
```

```
plot(PaPiRings1, series = "AFO1001a")

plot(PaPiRings, series = "AFO1001a", years = c(1990, 2000))
plot(PaPiRings$AFO1001a)
```

---

plotRings *Plot xRing Objects*

Description

Plot "xRing" objects.

Usage

```
plotRings(x, xlim = NULL, ylim = NULL, id = NULL, corr = NULL, EwLw = TRUE)
```

Arguments

x	an object of class "xRing"
xlim	the x limits of the plot. The default value, NULL, indicates that the whole profile will be plotted.
ylim	the y limits of the plot.
id	a sufix to be added to the name of the series (<series_name> [id])
corr	value to be print at the top of the graph
EwLw	logical. If TRUE the earlywood and latewood assignments are plotted, by default is TRUE

Value

None. A plot is produced.

See Also

plot.xRing

Examples

```
if (interactive()) {
  data(PaPiRaw)
  data(PaPiSpan)

  PaPi <- detectRings(PaPiRaw[, 1, drop = FALSE], PaPiSpan)
  plotRings(PaPi$AFO1001a)
  plotRings(PaPi, series = "AFO1001a")
  plotRings(PaPi, series = "AFO1001a", xlim = c(120, 450))
```

```
  PaPi1 <- detectEwLw(PaPi, ew = 0.5)
  plotRings(PaPi1, series = "AFO1001a", EwLw = FALSE)
  plotRings(PaPi1, series = "AFO1001a")
}
```

---

print *Print xRing and xRingList Objects*

Description

Print method for objects of class "xRing" and "xRingList".

Usage

```
## S3 method for class 'xRing'
print(x, ...)

## S3 method for class 'xRingList'
print(x, ...)
```

Arguments

x	the object of class "xRing" or "xRingList" to print
...	additional parameters

Value

None

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi <- detectRings(PaPiRaw, PaPiSpan)
class(PaPi)
print(PaPi$AFO1001a)
PaPi$AFO1001a
PaPi$AFO1001a[]
print(PaPi)
PaPi
```

removeRing *Remove Tree-Ring Border(s)*

Description

Remove the closest tree-ring border

Usage

```
removeRing(object, x, series = NULL)
```

Arguments

object	an object of class "xRing" or "xRingList"
x	the position to delete the closest tree-ring border
series	the name of the series to be changed when the object is a "xRingList", by default is NULL

Value

an object of class "xRing" or "xRingList" without the tree-ring border at the position x for the series given by series argument

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi <- detectRings(PaPiRaw, PaPiSpan)
plotRings(PaPi$AFO1001a)
abline(v = 60, lty = 2, col = 2)
PaPi$AFO1001a <- removeRing(PaPi$AFO1001a, x = 60)
# PaPi$AFO1001a <- removeRing(PaPi$AFO1001a, x = locator(1)$x)
plotRings(PaPi$AFO1001a)
```

selectProfiles *Select Profile(s)*

Description

Uses a line to select a profile (or a region of interest), when selecting a radius the line should start at the pith side and end at the bark side of the sample.

Usage

```
selectProfiles(im, nPixel = 50, cal = NULL, multiple = TRUE)
```

Arguments

im	an image
nPixel	the width of the line
cal	calibration
multiple	a single or several profiles

Value

a vector with the average grayvalue along the selected line when a multiple is TRUE and a list when multiple is FALSE

Examples

```
if (interactive()) {
  # read a sample file
  im <- imRead(file = system.file("img", "AFO1046.1200dpi.png", package = "xRing"))

  # to display the image
  imDisplay(im)

  # select a profile
  profile <- selectProfile(im)

  # to display the profile
  plot(profile, type = "l")
}
```

---

setLastYear *Set Last Year*

Description

Changes the calendar year of the last ring for a specific series.

Usage

```
setLastYear(x, lastYear, series = NULL)
```

Arguments

x	an "xRing" or "xRingList" object
lastYear	the new calendar year for the last tree ring
series	individual series to be changed when the object is a "xRingList", by default is NULL

Value

the modified input object with new set last ring of the specified series.

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi <- detectRings(PaPiRaw, PaPiSpan)
plot(PaPi, series = "AFO1001b")
PaPi <- setLastYear(PaPi, 2005, series = "AFO1001b")
plot(PaPi, series = "AFO1001b")
```

---

stepIncrease *Calculate the Steps Thickness of the Calibration Wedge*

Description

convenience function to calculate the thickness of each steps of the calibration wedge for wedges with continous step increase.

Usage

```
stepIncrease(step.increase = 0.24, nsteps = 7)
```

Arguments

step.increase	height increase per wedge step
nsteps	total number of steps (the first step has the thickness of 0 - the area beside the wedge. Mention that when setting nsteps)

Value

a numeric vector

toxRing *Create an "xRing" Object*

Description

Converts a dataframe with X-ray microdensity profiles into an "xRing" object

Usage

```
toxRing(x, y = NULL, seriesName)
```

Arguments

x	a dataframe with X-ray microdensity profiles
y	a dataframe with the numerical values of the first and last year in columns. The individual series are specified as row names.
seriesName	the name of series from x and y to be used to produce the "xRing" object.

Value

an "xRing" object, an S3 class with the following elements:

`profile.raw` a `vector` with the input density profile

`span` first and last year

`name` a `string` giving the series name

See Also

toxRingList

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi.AFO1001a <- toxRing(PaPiRaw, PaPiSpan, seriesName = "AFO1001a")
class(PaPi.AFO1001a)
```

`toxRingList` *Create a "xRingList" Object*

Description

Converts a dataframe with X-ray microdensity profiles in an "xRingList" object

Usage

```
toxRingList(x, y = NULL)
```

Arguments

`x`	a dataframe with X-ray microdensity profiles
`y`	a dataframe with the numerical values of the first and last year in columns. The individual series are specified as row names. By default is NULL

Value

an "xRingList" object, an S3 class which list membera are "xRing" objects containing:

`profile.raw` a `vector` with the input density profile

`span` first and last year

`name` a `string` giving the series name

See Also

toxRing

Examples

```
data(PaPiRaw)
data(PaPiSpan)
PaPi <- toxRingList(PaPiRaw, PaPiSpan)
class(PaPi)
```

Index