Package 'MtreeRing'

October 16, 2018

Type Package

Title A Shiny Application for Automatic Measurements of Tree-Ring Widths on Digital Images

Version 1.1

Author Jingning Shi, Wei Xiang

Maintainer Jingning Shi <snow940220@bjfu.edu.cn>

Depends R (>= 3.3.0), magrittr (>= 1.5)

Imports png, jpeg, tiff, bmp, magick, imager, dplR, spatstat, measuRing, shiny, shinydashboard, shinyWidgets

Description

Use morphological image processing and edge detection algorithms to automatically identify tree-ring boundaries on digital images. Tree-ring boundaries are determined by the changes of light reflectance from late wood to early wood. Two geometric models are provided to calibrate the errors resulted from inclined rings. The package provides a Shiny-based application, allowing R beginners to easily analyze tree-ring images and export ring-width series in standard file formats.

License GPL-3

LazyData TRUE

NeedsCompilation no

Repository CRAN

Encoding UTF-8

Date/Publication 2018-10-16 18:00:03 UTC

R topics documented:

`MtreeRing-package` *A Shiny Application for Automatic Measurements of Tree-Ring Widths on Digital Images*

Description

Use morphological image processing and edge detection algorithms to automatically identify tree-ring boundaries on digital images. Tree-ring boundaries are determined by the changes of light reflectance from late wood to early wood. Two geometric models are provided to calibrate the errors resulted from inclined rings. The package provides a Shiny-based application, allowing R beginners to easily analyze tree-ring images and export ring-width series in standard file formats.

Details

Package:	MtreeRing
Type:	Package
License:	GPL-3
Maintainer:	Jingning Shi <snow940220@bjfu.edu.cn>
LazyData:	TRUE

Author(s)

Jingning Shi, Wei Xiang

`autoDetect` *Automatic detection of tree-ring boundaries*

Description

This function is used to automatically detect tree-ring boundaries along the user-defined path.

Usage

```
autoDetect(ring.data, seg = 1, auto.path = TRUE, manual = FALSE,
    method = "watershed", incline = FALSE, sample.yr = NULL,
    watershed.threshold = "auto", watershed.adjust = 0.8,
    struc.ele1 = NULL, struc.ele2 = NULL, marker.correct = FALSE,
    default.canny = TRUE, canny.t1, canny.t2, canny.smoothing = 2,
    canny.adjust = 1.4, path.dis = 1, origin = 0, border.color = "black",
    border.type = 16, label.color = "black", label.cex = 1.2)
```

Arguments

ring.data	A magick image object produced by imgInput.
seg	An integer specifying the number of image segments.
auto.path	A logical value. If TRUE, a path is automatically created at the center of the image. If FALSE, the function allows the user to create a sub-image and a path by interactive clickings. See details below.
manual	A logical value indicating whether to skip the automatic detection. If TRUE, ring boundaries are visually identified using the function visualSelect.
method	A character string specifying how ring borders are detected. It requires one of the following characters: "watershed", "canny", or "lineardetect". See details below.
incline	A logical value indicating whether to correct errors due to inclined rings. If TRUE, two horizontal paths will be added to the image.
sample.yr	NULL or an integer giving the year of formation of the left-most ring. If NULL, use the current year.
watershed.threshold	The threshold used for producing the marker image, either a numeric from 0 to 1, or the character "auto" (using the Otsu algorithm), or a character of the form "XX%" (e.g., "58%").
watershed.adjust	A numeric used to adjust the Otsu threshold. The default is 1 which means that the threshold will not be adjusted. The area of early-wood regions in the marker image will reduce along with the decrease of watershed.adjust.
struc.ele1	NULL or a vector of length two specifying the width and height of the first structuring element. If NULL, the size of the structuring element is determined according to the argument dpi.
struc.ele2	NULL or a vector of length two specifying the width and height of the second structuring element. If NULL, the size of the structuring element is determined according to the argument dpi.
marker.correct	A logical value indicating whether to relabel early-wood regions by comparing the values of their left-side neighbours.
default.canny	A logical value. If TRUE, upper and lower Canny thresholds are determined automatically.
canny.t1	A numeric giving the threshold for weak edges.
canny.t2	A numeric giving the threshold for strong edges.
canny.smoothing	An integer specifying the degree of smoothing.
canny.adjust	A numeric used as a sensitivity control factor for the Canny edge detector. The default is 1 which means that the sensitivity will not be adjusted. The number of detected borders will reduce along with the increase of this value.
path.dis	A numeric specifying the perpendicular distance between two paths when the argument incline = TRUE. The unit is in mm.

origin	A numeric specifying the origin in smoothed gray to find the ring borders. See ringBorders in the package measuRing.
border.color	Color for ring borders.
border.type	Symbol for ring borders. See pch in points for possible values and their shapes.
label.color	Color for years and ring numbers.
label.cex	The magnification to be used for years and ring numbers.

Details

If auto.path = TRUE, the user can create a user-defined rectangular sub-image and a horizontal path by interactively clicking on the tree-ring image. The automatic detection will be performed within the rectangular sub-image along a pre-determined path. To create the sub-image and the path, follow these steps.

- Step 1. Select the left and right edges of the rectangle
  If partial.rings = TRUE, the user can point the mouse at any desired locations and click the left mouse button to add each edge. If partial.rings = FALSE, the left and right boundaries of the original image will be used directly as the left and right edges of the rectangle (i.e., skip this step).
- Step 2. Select the top and bottom edges of the rectangle
  The user can point the mouse at any desired locations and click the left mouse button to add each edge. The width of the rectangle is defined as the distance between the top and bottom edges, and should not be unnecessarily large to reduce time consumption and memory usage. Creating a long and narrow rectangle if possible.
- Step 3. Create a path
  After creating the rectangular sub-image, the user can add a horizontal path by left-clicking on the sub-image (generally in the middle of the sub-image, try to choose a clean defect-free area). Ring borders and other markers are plotted along this path. If incline = TRUE, two paths are added simultaneously.

After creating the sub-image and the path, this function will open several graphical windows and plot detected ring borders on image segments. The number of image segments is controlled by the argument seg (see above).

Argument method determines how ring borders are identified.

- If method = "watershed", this function uses the watershed algorithm to obtain ring borders (Soille and Misson, 2001).
- If method = "canny", this function uses the Canny algorithm to detect borders.
- If method = "lineardetect", a linear detection algorithm from the measuRing package is used to identify ring borders (Lara et al., 2015). Note that incline = TRUE is not supported in this mode and path will be created automatically in the middle position of the image without the need to specify by the user.

If the argument method = "watershed" or "canny", the original image is processed by morphological openings and closings using rectangular structuring elements of increasing size before detecting borders. The first small structuring element is used to remove smaller dark spots in early-wood regions and the second large structuring element is used to remove light strips in late-wood regions.

Value

A matrix (grayscale image) or array (color image) representing the tree-ring image.

Note

This function uses the function `locator` to record mouse positions so it only works on "X11", "windows" and "quartz".

Author(s)

Jingning Shi

References

Soille, P., Misson, L. (2001) Tree ring area measurements using morphological image analysis. *Canadian Journal of Forest Research* **31**, 1074-1083. doi: 10.1139/cjfr-31-6-1074

Lara, W., Bravo, F., Sierra, C.A. (2015) measuRing: An R package to measure tree-ring widths from scanned images. *Dendrochronologia* **34**, 43-50. doi: 10.1016/j.dendro.2015.04.002

Examples

```
## Find the image file name in package MtreeRing:
img.name <- system.file("001.png", package = "MtreeRing")

## Read and plot the image:
t1 <- imgInput(img = img.name, dpi = 1200)

## Split a long core sample into 3 pieces to
## get better display performance and use the
## watershed algorithm to detect ring borders:
t2 <- autoDetect(ring.data = t1, seg = 3, method = 'watershed')
```

---

calcRingWidth *Generate a ring-width series*

Description

This function can calculate the ring-width series according to detected ring borders.

Usage

```
calcRingWidth(ring.data, seriesID)
```

Arguments

`ring.data`	A matrix or array produced by `autoDetect` or `visualSelect`.
`seriesID`	A character string specifying the column name of the ring-width series.

Value

A data frame. The series ID is the column name and years are row names.

Author(s)

Jingning Shi

Examples

```
## Find the image file name in package MtreeRing:
img.name <- system.file("001.png", package = "MtreeRing")

## Read and plot the image:
t1 <- imgInput(img = img.name, dpi = 1200)

## Split a long core sample into 3 pieces to
## get better display performance and use the
## watershed algorithm to detect ring borders:
t2 <- autoDetect(ring.data = t1, seg = 3, method = 'watershed')

## Calculate ring widths from the attribute list of t2:
rw.df <- calcRingWidth(ring.data = t2, seriesID = "940220")
```

---

imgInput *Read and plot a tree-ring image file*

Description

This function can read an image file from the hard disk and plot it in a newly-opened graphical device.

Usage

```
imgInput(img, dpi = NULL, rotate = 0,
    RGB = c(0.299, 0.587, 0.114), magick = TRUE)
```

Arguments

img	A character string indicating the path of the image file. Supported formats include png, tiff, jpg and bmp.
dpi	An integer specifying the dpi of the image file. A minimum of 300 dpi is required when running automatic detection.
rotate	An integer specifying how many degrees to rotate (clockwise). It requires one of the following values: 0, 90, 180 or 270.
RGB	A numeric vector of length 3 giving the weight of RGB color channels.
magick	A logical value. If TRUE, the package magick is used to read images whose file sizes are over 10MB.

Details

Proper image preparation has a great influence on the measurement of ring widths. A tree-ring image should not contain irrelevant or redundant features such as wooden mounts where cores are glued. The larger the file size of an image, the slower the image processing operation will be.

Pith side of a tree-ring sample should be placed on the right side of the graphical window. Use the argument `rotate` to change its position.

It is highly recommended to use the default value `magick = TRUE`, because the package `magick` can significantly reduce the memory usage.

Value

A magick image object containing the image data.

Author(s)

Jingning Shi

Examples

```
## Find the image file name in package MtreeRing:
img.name <- system.file("001.png", package = "MtreeRing")

## Read and plot the image:
t1 <- imgInput(img = img.name, dpi = 1200)
```

---

launchMtRApp *Run Shiny-based Application*

Description

Run a Shiny-based application within the system's default web browser. The application provides a beginner-friendly graphical interface and supports more flexible mouse-based interactions.

Usage

```
launchMtRApp()
```

Author(s)

Jingning Shi, Wei Xiang

nearPith *Calibrate ring-width series*

Description

This function can calibrate the ring-width series using arcs of inner rings.

Usage

```
nearPith(ring.data, inner.arc = TRUE, last.yr = NULL,
      color = 'black', border.type = 16, label.cex = 1.5)
```

Arguments

ring.data	A magick image object produced by imgInput.
inner.arc	A logical value indicating whether to calibrate the ring-width series using the arcs of inner rings. See details
last.yr	NULL or an integer giving the year of formation of the left-most ring. If NULL, index numbers (starting from 1) is used instead of the year.
color	Color for labels
border.type	Symbol for ring borders. See pch in points for possible values and their shapes.
label.cex	The magnification to be used for years and ring numbers.

Details

This function allows the user to create a path and visually identify ring borders by clicking on the graphical window generated by imgInput.

- If inner.arc = TRUE, the ring-width series is calibrated using arcs of inner rings (Duncan, 1989).

  First, the user can click the left mouse button to add a horizontal path. The path should traverse an appropriate arc (read the reference below for more details). Then, the user can add three points to the selected arc by left-clicking. The first point should be placed on the left endpoint of the arc and the second point is placed on the right endpoint.

  After adding two points, a vertical dashed line will be plotted automatically according to the (x,y) positions of endpoints. The third points should be placed on the intersection of the vertical dashed line and the selected arc.

  Finally, the user is prompted to visually mark ring borders along the path. The termination of visual selection is similar to visualSelect. Note that the left endpoint of the arc will be considered as the last ring border on the path without the need to mark it.

  The ring-width series are corrected using formulas proposed by Duncan (1989).

- If inner.arc = FALSE, the user can create a path following the direction of pith.

  First, the user can add two points by left-clicking on the image. A path passing through these two points will be plotted. The path should be selected along the rays from bark to pith.

  Then, the user can visually mark ring borders along the path. The termination of visual selection is similar to visualSelect.

Value

A data frame containing the calibrated ring-width series.

Author(s)

Jingning Shi

References

Duncan R. (1989) An evaluation of errors in tree age estimates based on increment cores in Kahikatea (Dacrycarpus dacrydiodes). *New Zealand Natural Sciences* **16(4)**, 1-37.

Examples

```
## Find the image file name in package MtreeRing:
img.name <- system.file("arc.png", package = "MtreeRing")

## Read and plot the image:
t1 <- imgInput(img = img.name, dpi = 1200)

## Use the arcs of inner rings to calibrate ring-width series:
t2 <- nearPith(ring.data = t1, inner.arc = TRUE, last.yr = 2016)


## Try another method to measure ring widths:
t3 <- nearPith(ring.data = t1, inner.arc = FALSE, last.yr = 2016)
```

---

visualSelect *Edit ring borders visually*

Description

This function can delete existing ring borders and add new borders visually.

Usage

```
visualSelect(ring.data, del = NULL, del.u = NULL, del.l = NULL, add = FALSE)
```

Arguments

ring.data	A matrix or array produced by autoDetect.
del	A numeric vector giving the border numbers to be deleted.
del.u	A numeric vector giving the border numbers to be deleted on the upper path.
del.l	A numeric vector giving the border numbers to be deleted on the lower path.
add	A logical value indicating whether to re-add ring borders visually.

Details

This function is used to delete existing ring borders or add new borders by interactively clicking on the image segments.

If the user creates only one path (`incline = FALSE`), the argument `del` is used to delete ring borders. If the user creates two paths (`incline = TRUE`), arguments `del.u` and `del.l` are used to delete ring borders.

If `add = TRUE`, graphical windows where image segments are plotted will be activated sequentially. When one graphical window is activated, the user can add new borders by left-clicking the mouse along the path. This visual selection process can be terminated by clicking the right button and selecting 'Stop' from the menu, or from the 'Stop' button on the top-left corner of the graphical window (depending on the R version).

Once the user terminates the visual selection process on one segment, the current graphical window will be closed automatically, and the graphical window of the following segment is activated. When all graphical windows are closed, this function will re-open graphical windows and plot new borders.

This function can perform both operations (deletion and addition) in one call. A deletion of borders takes precedence over addition.

Value

A matrix or array representing the user-selected subset of the image.

Author(s)

Jingning Shi

Examples

```
## Find the image file name in package MtreeRing:
img.name <- system.file("001.png", package = "MtreeRing")

## Read and plot the image:
t1 <- imgInput(img = img.name, dpi = 1200)

## Split a long core sample into 3 pieces to
## get better display performance and use the
## watershed algorithm to detect ring borders:
t2 <- autoDetect(ring.data = t1, seg = 3, method = 'watershed')

## Not modify t2, create a new array object t3. Delete
## some borders without adding new borders:
t3 <- visualSelect(ring.data = t2, del = c(1, 3, 5, 19:21), add = FALSE)
```

Index