Package 'archiDART'

May 11, 2017

Type Package

Title Plant Root System Architecture Analysis Using DART and RSML Files

Version 2.0

Date 2017-05-11

Author Benjamin M Delory, Caroline Baudson, Yves Brostaux, Guillaume Lobet, Patrick du Jardin, Loic Pages, Pierre Delaplace

Maintainer Benjamin M Delory <Benjamin.Delory@leuphana.de>

Description Analysis of complex plant root system architectures (RSA) using the output files created by Data Analysis of Root Tracings (DART), an open-access software dedicated to the study of plant root architecture and development across time series (Le Bot et al (2010) ``DART: a software to analyse root system architecture and development from captured images'', Plant and Soil, <DOI:10.1007/s11104-009-0005-2>), and RSA data encoded with the Root System Markup Language (RSML) (Lobet et al (2015) ``Root System Markup Language: toward a unified root architecture description language'', Plant Physiology, <DOI:10.1104/pp.114.253625>). More information can be found in Delory et al (2016) ``archiDART: an R package for the automated computation of plant root architectural traits'', Plant and Soil, <DOI:10.1007/s11104-015-2673-4>.

License GPL-2

Imports stats, utils, graphics, grDevices, XML, rgl

NeedsCompilation no

Repository CRAN

Date/Publication 2017-05-11 07:40:23 UTC

R topics documented:

`archidraw` *Plotting Vectorized Root Systems*

Description

Plotting vectorized root systems for selected observation dates using Data Analysis of Root Tracings (DART) output files and RSA data encoded with the Root System Markup Language (RSML). More information can be found in *Delory et al (2016)*, *Le Bot et al (2010)*, and *Lobet et al (2015)*.

Usage

```
archidraw(inputlie=NULL, inputrsml=NULL, res=NULL, unitlength="px", rsml.connect=FALSE,
        rsml.date=NULL, unitangle="d", rotation=0, numdate=NULL, finalscale=NULL,
        coldate=par("col"), main=NULL, xlab=NULL, ylab=NULL, zlab=NULL, xlim=NULL,
          ylim=NULL, zlim=NULL,...)
```

Arguments

`inputlie` A character string specifying the path to the folder containing the lie files created by DART. This argument could be the same as `inputrsml`.

`inputrsml` A character string specifying the path to the folder containing the RSML files. This argument could be the same as `inputlie`.

`res` Mandatory for DART files only. **If images were acquired with a flatbed scanner**: a numeric value specifying the resolution of the images used to vectorize the root systems with DART (resolution is expressed in dots/inch). **If images were acquired with a digital camera**: the ratio between the length of a reference object located on the image expressed in pixels and the actual length of the same object expressed in inches. For DART files, this argument must be specified if `unitlength="mm"` or `unitlength="cm"`. For RSML files, the resolution is imported from the metadata and does not need to be specified as an argument.

`unitlength` A character string specifying the unit of length that must be used to plot the root systems. The value acceptable for this argument could be either “px” for pixels, “cm” for centimetres or “mm” for millimetres.

`rsml.connect` Only used for RSML files. A logical value that must be specified when `is.null(inputrsml)=FALSE`. If `TRUE`, the lateral roots are connected to their parent root (see details). If `FALSE`, no connection is made between daughter and mother roots.

`rsml.date` Only used for RSML files. A numeric value specifying the root system age (the unit of time is defined by the user). If `rsml.date=NULL`, a default value of 1 is used by the function. If the root system age is stored as a continuous function along the root segments, a character string specifying the name/label of the function can be used (see details).

`unitangle` A character string specifying the unit that must be used to express the calculated angles. The value acceptable for this argument could be either “d” for degrees (default value) or “r” for radians.

`rotation` A numeric value specifying the angle (expressed in `unitangle`) that must be used to perform a clockwise rotation of 2D root systems (see details).

`numdate` A numeric value or a vector of numeric values specifying the identification number of all observation dates at which the root system(s) must be plotted (see details).

`finalscale` A logical value that must be specified when `is.null(numdate)=FALSE`. If `TRUE`, the root system architecture at the selected observation dates is plotted with the same axis scales to that used to draw the root system architecture at the last observation date. If `FALSE`, the root system architecture at each observation date is plotted using different axis ranges of coordinates (`xlim` and `ylim`).

`coldate` A vector specifying the color(s) that must be used to plot the root system(s) at the selected observation dates (see details).

`main` A character string giving a main title for the plot. When `main=NULL`, the default title for each plot is the name of the corresponding lie/RSML file.

`xlab` A character string giving a label for the X axis. When `xlab=NULL`, the default label for the X axis is X (`unitlength`).

`ylab` A character string giving a label for the Y axis. When `ylab=NULL`, the default label for the Y axis is Y (`unitlength`).

`zlab` A character string giving a label for the Z axis. When `zlab=NULL`, the default label for the Z axis is Z (`unitlength`).

`xlim` A numeric vector (length=2) giving the X limits of the plot using the same unit as `unitlength`.

`ylim` A numeric vector (length=2) giving the Y limits of the plot using the same unit as `unitlength`.

`zlim` A numeric vector (length=2) giving the Z limits of the plot using the same unit as `unitlength`.

`...` Additional graphical parameters (see details).

Details

To run `archidraw` efficiently, DART and RSML files must have been saved with their appropriate extension (.lie or .rsml). By default (when `numdate=NULL`), only the root system architecture at the last observation date is plotted.

If the number of elements specified in `coldate` is lower than the number of observation dates, `archidraw` replicates the elements stored in `coldate` to plot the root system(s). When the number of elements specified in `coldate` is greater than the number of observation dates, only the first corresponding elements in `coldate` are used by `archidraw` to plot the root system(s).

After reading the RSML files located in `inputrsml`, the data associated with each root system are converted into a data frame possessing the same structure as the lie files created by DART. The `archidraw` function then uses these data frames to plot the root system(s).

The argument `rsml.date` is used to create a tps file for each root system located in `inputrsml`. If the root system age is stored as a continuous function along the root segments, the `rsml.date` argument can be a character string specifying the name/label of the function. For instance, the root system age at the first apparition of each node is stored as a continuous function called “age” in

RSML files containing RSA data simulated with the ArchiSimple model (*Pages et al (2013)*). The “age” function attaches the root system age along the root segments, so that each node is associated with one age value. An RSML library containing 3D RSA data simulated with ArchiSimple is available here: `https://doi.org/10.5281/zenodo.208214`.

The `rsml.connect` argument can be used to connect the lateral roots to their corresponding mother root. If `rsml.connect=TRUE`, each point starting a lateral root is connected to the nearest point located on its mother root. DBase is approximated by the distance between the parent root base and the point located on the mother root that is closest to the point starting a lateral root. Using RSML files, only RSA data associated with roots possessing a branching order lower or equal to 5 are used for the computation of RSA traits.

Due to technical limitations, it is sometimes easier to take a picture of a root system if it is rotated before image acquisition. In this case, the vectorized root system depicts a rotation compared with the natural plant root system. In order to make a correction, one can use `rotation` to specify an angle value expressed in `unitangle` to rotate the vectorized root system clockwise before plotting.

Additional graphical parameters can be used to personalize the graphical outputs (see `par`).

Value

Returns a plot for each vectorized root system located in `inputlie/inputrsml`.

Author(s)

Benjamin M. Delory, Caroline Baudson, Yves Brostaux, Guillaume Lobet, Loic Pages, Patrick du Jardin, Pierre Delaplace

References

Delory B.M., Baudson C., Brostaux Y., Lobet G., du Jardin P., Pages L., Delaplace P. (2016) archiDART: an R package for the automated computation of plant root architectural traits, *Plant and Soil*, DOI: 10.1007/s11104-015-2673-4.

Le Bot J., Serra V., Fabre J., Draye X., Adamowicz S., Pages L. (2010) DART: a software to analyse root system architecture and development from captured images, *Plant and Soil*, DOI: 10.1007/s11104-009-0005-2.

Lobet G., Pound M.P., Diener J., Pradal C., Draye X., Godin C., Javaux M., Leitner D., Meunier F., Nacry P., Pridmore T.P., Schnepf A. (2015) Root System Markup Language: Toward a Unified Root Architecture Description Language, *Plant Physiology*, DOI: 10.1104/pp.114.253625.

Pages L., Becel C., Boukcim H., Moreau D., Nguyen C., Voisin, A-S. (2013) Calibration and evaluation of ArchiSimple, a simple model of root system architecture, *Ecological Modelling*, DOI: 10.1016/j.ecolmodel.2013.11.014.

See Also

`par`.

Examples

```
## Locate folder with DART and RSML files
path <- system.file("extdata", package="archiDART")

##-----------------
## DART files only
##-----------------

## Plot RSA at several observation dates
## One color for each observation date
archidraw(inputlie=path, res=75, unitlength="cm", numdate=c(15,31), finalscale=TRUE,
coldate=rainbow(31), lwd=2,las=1, bty="l", asp=1, xaxp=c(0,30,3), yaxp=c(0,90,9))

##-----------------
## RSML files only
##-----------------

## Plot RSA at the last observation date
## One color for each observation date
archidraw(inputrsml=path, unitlength="cm", rsml.date="age", coldate=rainbow(16), lwd=2)

##---------------------
## DART and RSML files
##---------------------

archidraw(inputlie=path, inputrsml=path, res=75, unitlength="cm", rsml.date="age",
coldate="black", lwd=2, las=1, bty="l", asp=1)
```

archigrow *Computing Growth Rates and Plotting of Vectorized Root Systems*

Description

Calculation of growth rate matrices and plotting of vectorized root systems for selected observation dates using Data Analysis of Root Tracings (DART) output files and RSA data encoded with the Root System Markup Language (RSML). In the final plots, the color code used for each link constituting a vectorized root system depends on its corresponding growth rate value. More information can be found in *Delory et al (2016)*, *Le Bot et al (2010)*, and *Lobet et al (2015)*.

Usage

```
archigrow(inputlie=NULL, inputtps=NULL, inputrsml=NULL, res=NULL, unitlength="px",
          rsml.date="age", rsml.connect=TRUE, plot=TRUE, export.colors=FALSE,
        unittime=NULL, unitangle="d", rotation=0, numdate=NULL, finalscale=NULL,
         coldyn=NULL, GRscale=NULL, main=NULL, xlab=NULL, ylab=NULL, zlab=NULL,
          xlim=NULL, ylim=NULL, zlim=NULL, ...)
```

Arguments

`inputlie` A character string specifying the path to the folder containing the lie files created by DART. This argument could be the same as `inputtps`/`inputrsml`.

`inputtps` A character string specifying the path to the folder containing the tps files created by DART. This argument could be the same as `inputlie`/`inputrsml`.

`inputrsml` A character string specifying the path to the folder containing the RSML files. This argument could be the same as `inputlie`/`inputtps`.

`res` Mandatory for DART files only. **If images were acquired with a flatbed scanner**: a numeric value specifying the resolution of the images used to vectorize the root systems with DART (resolution is expressed in dots/inch). **If images were acquired with a digital camera**: the ratio between the length of a reference object located on the image expressed in pixels and the actual length of the same object expressed in inches. For DART files, this argument must be specified if `unitlength="mm"` or `unitlength="cm"`. For RSML files, the resolution is imported from the metadata and does not need to be specified as an argument.

`unitlength` A character string specifying the unit of length that must be used by the function to plot the root systems. The value acceptable for this argument could be either “px” for pixels, “cm” for centimetres or “mm” for millimetres.

`rsml.date` Only used for RSML files. A numeric value specifying the root system age (the unit of time is defined by the user). If `rsml.date=NULL`, a default value of 1 is used by the function. If the root system age is stored as a continuous function along the root segments, a character string specifying the name/label of the function can be used (see details).

`rsml.connect` Only used for RSML files. A logical value that must be specified when `is.null(inputrsml)=FALSE`. If `TRUE`, the lateral roots are connected to their parent root (see details). If `FALSE`, no connection is made between daughter and mother roots.

`plot` A logical value. Should the root system(s) be plotted? `TRUE` is yes, `FALSE` is no.

`export.colors` A logical value. Should the color values be exported? If `TRUE`, a matrix giving the color code of each root at each observation date is exported. If `FALSE`, no color matrix is exported.

`unittime` A character string specifying the unit of time used to express the data (all character strings are valid for this argument). For RSML files, unittime is imported from the metadata and does not need to be specified as an argument.

`unitangle` A character string specifying the unit that must be used by the function to express the calculated angles. The value acceptable for this argument could be either “d” for degrees (default value) or “r” for radians.

`rotation` A numeric value specifying the angle (expressed in `unitangle`) that must be used to perform a clockwise rotation of 2D root systems (see details).

`numdate` A numeric value or a vector of numeric values specifying the identification number of all observation dates at which the root system(s) must be plotted (see details).

`finalscale` A logical value that must be specified when `is.null(numdate)=FALSE`. If `TRUE`, the root system architecture at the selected observation dates is plotted with the same axis scales to that used to draw the root system architecture at the last observation date. If `FALSE`, the root system architecture at each observation date is plotted using different axis ranges of coordinates (`xlim` and `ylim`).

`coldyn` A vector specifying the color(s) that must be used to plot the root system(s) at the selected observation dates. The function colors each link constituting a vectorized root system depending on its corresponding growth rate value. To do so, `archigrow` interpolates the color(s) contained in `coldyn` between the minimum and the maximum values of the computed standardized growth rate matrix (if `is.null(GRscale)=TRUE`) or between `min(GRscale)` and `max(GRscale)` (if `is.null(GRscale)=FALSE`).

`GRscale` A numeric vector (length=2) specifying the minimum and the maximum growth rate values (expressed in `unitlength`/`unittime`) that must be used to plot each vectorized root system.

`main` A character string giving a main title for the plot. When `is.null(main)=TRUE`, the default title for each plot is the name of the corresponding lie/RSML file.

`xlab` A character string giving a label for the X axis. When `is.null(xlab)=TRUE`, the default label for the X axis is X (`unitlength`).

`ylab` A character string giving a label for the Y axis. When `is.null(ylab)=TRUE`, the default label for the Y axis is Y (`unitlength`).

`zlab` A character string giving a label for the Z axis. When `is.null(zlab)=TRUE`, the default label for the Z axis is Z (`unitlength`).

`xlim` A numeric vector (length=2) giving the X limits of the plot using the same unit as `unitlength`.

`ylim` A numeric vector (length=2) giving the Y limits of the plot using the same unit as `unitlength`.

`zlim` A numeric vector (length=2) giving the Z limits of the plot using the same unit as `unitlength`.

`...` Additional graphical parameters (see details)

Details

To run `archigrow` efficiently, DART and RSML files must have been saved with their appropriate extension (.lie, .tps, or .rsml). By default (when `numdate=NULL`), only the root system architecture at the last observation date is plotted.

If `inputtps` contains more than one tps file, the number of tps files in `inputtps` must be equal to the number of lie files in `inputlie` and corresponding lie and tps files must have the same name.

After reading the RSML files located in `inputrsml`, the data associated with each root system are converted into a data frame possessing the same structure as the lie files created by DART. The `archigrow` function then uses these data frames to plot the root system(s).

The argument `rsml.date` is used to create a tps file for each root system located in `inputrsml`. If the root system age is stored as a continuous function along the root segments, the `rsml.date` argument can be a character string specifying the name/label of the function. For instance, the root system age at the first apparition of each node is stored as a continuous function called “age” in RSML files containing RSA data simulated with the ArchiSimple model (*Pages et al (2013)*). The “age” function attaches the root system age along the root segments, so that each node is associated with one age value. An RSML library containing 3D RSA data simulated with ArchiSimple is available here: https://doi.org/10.5281/zenodo.208214.

The `rsml.connect` argument can be used to connect the lateral roots to their corresponding mother root. If `rsml.connect=TRUE`, each point starting a lateral root is connected to the nearest point located on its mother root. DBase is approximated by the distance between the parent root base and the point located on the mother root that is closest to the point starting a lateral root. Using RSML files, only RSA data associated with roots possessing a branching order lower or equal to 5 are used for the computation of RSA traits.

Due to technical limitations, it is sometimes easier to take a picture of a root system if it is rotated before image acquisition. In this case, the vectorized root system depicts a rotation compared with the natural plant root system. In order to make a correction, one can use `rotation` to specify an angle value expressed in `unitangle` to rotate the vectorized root system clockwise before plotting.

Additional graphical parameters can be used to personalize the graphical outputs (see `par`).

Growth rates are calculated as follows:

- For the first observation date, it is calculated as the ratio of the root length to the root system age.
- For other observation dates (t), it is calculated as the difference between the root length at time t and t-1 divided by the difference between the root system age at time t and t-1.

Value

Returns a list of secondary lists. Each element of the primary list is named as its corresponding lie/RSML file. For each lie/RSML file, the secondary list contains the following items:

`unit` The unit of length/time used to express the results.

`time` A data frame containing the following columns: **Date** (the identification number of each observation date) and **Age** (the root system age).

`GR` A matrix containing the growth rate values computed by the function. The matrix contains the following columns: **Root** (the identification number of each root constituting a vectorized root system), **Ord** (the branching order; only for DART files or for RSML files with `rsml.connect=TRUE`), **DateX** (the growth rate value of each root at the observation date X).

`colors` Only if `export.colors=TRUE`. A matrix containing the colors used to plot the root system. The matrix contains the following columns: **Root** (the identification number of each root constituting a vectorized root system), **DateX** (the color used to plot each root at the observation date X).

If `plot=TRUE`, returns a plot for each vectorized root system located in `inputlie/inputrsml`.

Author(s)

Benjamin M. Delory, Caroline Baudson, Yves Brostaux, Guillaume Lobet, Loic Pages, Patrick du Jardin, Pierre Delaplace

References

Delory B.M., Baudson C., Brostaux Y., Lobet G., du Jardin P., Pages L., Delaplace P. (2016) archiDART: an R package for the automated computation of plant root architectural traits, *Plant and Soil*, DOI: 10.1007/s11104-015-2673-4.

Le Bot J., Serra V., Fabre J., Draye X., Adamowicz S., Pages L. (2010) DART: a software to analyse root system architecture and development from captured images, *Plant and Soil*, DOI: 10.1007/s11104-009-0005-2.

Lobet G., Pound M.P., Diener J., Pradal C., Draye X., Godin C., Javaux M., Leitner D., Meunier F., Nacry P., Pridmore T.P., Schnepf A. (2015) Root System Markup Language: Toward a Unified Root Architecture Description Language, *Plant Physiology*, DOI: 10.1104/pp.114.253625.

Pages L., Becel C., Boukcim H., Moreau D., Nguyen C., Voisin, A-S. (2013) Calibration and evaluation of ArchiSimple, a simple model of root system architecture, *Ecological Modelling*, DOI: 10.1016/j.ecolmodel.2013.11.014.

See Also

`par`.

Examples

```
##---------------------------------------------
## Create a function (legendGR) to add a legend
##---------------------------------------------

legendGR<-function(list, coldyn, GRscale=NULL, raster=c(0.1,0.1,0.6,1), textx=0.8, l=10, ...){
start.col<-ncol(list$GR)-length(grep("Date", colnames(list$GR)))+1
end.col<-ncol(list$GR)
growthrate<-list$GR[,start.col:end.col]
colorlegend<-colorRampPalette(coldyn)(1000)
legendimage<-as.raster(matrix(rev(colorlegend), ncol=1))
par(mar=c(1,1,2,1))
plot(c(0,1),c(0,1),type="n", axes=FALSE, xlab="", ylab="", ...)
if (is.null(GRscale)==TRUE) {text(x=textx, y=seq(raster[2],raster[4],l=l),
labels=round(seq(round(min(growthrate),1),round(max(growthrate),1),l=l),1),...)}
else {text(x=textx, y=seq(raster[2],raster[4],l=l), labels=round(seq(round(min(GRscale),1),
round(max(GRscale),1),l=l),1),...)}
rasterImage(legendimage,xleft=raster[1],ybottom=raster[2],xright=raster[3],ytop=raster[4])}

## Locate folder with DART and RSML files
path <- system.file("extdata", package="archiDART")

##----------------
## DART files only
##----------------

layout(matrix(1:2,ncol=2),widths = c(3,1),heights = c(1,1))
par(mar=c(5,4,4,2)+0.1)
res1a <- archigrow(inputlie=path, inputtps=path, res=75, unittime="day",
unitlength="cm", plot=TRUE, export.colors=TRUE, coldyn=c("blue", "orange", "red"),
las=1, bty="l", asp=1, xaxp=c(0,30,3), lwd=2)
```

```
legendGR(list=res1a$ch7, coldyn=c("blue", "orange", "red"))

##--------------------
## DART and RSML files
##--------------------

layout(1)
par(mar=c(5,4,4,2)+0.1)
res1b <- archigrow(inputlie=path, inputtps=path, inputrsml=path, res=75, unittime="day",
unitlength="cm", rsml.connect=TRUE, rsml.date="age", plot=TRUE, export.colors=TRUE,
coldyn=c("black", "orange", "red"), las=1, bty="l", asp=1, xaxp=c(0,30,3), lwd=2)
unit<-res1b$`monocot-archisimple`$unit
legendGR(list=res1b$`monocot-archisimple`, coldyn=c("black", "orange", "red"),
main=paste("Growth rate monocot-archisimple (", unit, ")", sep=""), raster=c(0.2,0,0.7,1))
```

architect *Computing Traits Describing The Global Root System Architecture*

Description

Calculation of common root system architecture (RSA) traits from Data Analysis of Root Tracings (DART) output files and RSA data encoded with the Root System Markup Language (RSML). More information can be found in *Delory et al (2016)*, *Le Bot et al (2010)*, and *Lobet et al (2015)*.

Usage

```
architect(inputrac=NULL, inputtps=NULL, inputrsml=NULL, res=NULL, unitlength="px",
          rsml.date=NULL, rsml.connect=FALSE)
```

Arguments

`inputrac` A character string specifying the path to the folder containing the rac files created by DART. This argument could be the same as `inputtps`/`inputrsml`.

`inputtps` A character string specifying the path to the folder containing the tps files created by DART. This argument could be the same as `inputrac`/`inputrsml`.

`inputrsml` A character string specifying the path to the folder containing the RSML files. This argument could be the same as `inputrac`/`inputtps`.

`res` Mandatory for DART files only. **If images were acquired with a flatbed scanner**: a numeric value specifying the resolution of the images used to vectorize the root systems with DART (resolution is expressed in dots/inch). **If images were acquired with a digital camera**: the ratio between the length of a reference object located on the image expressed in pixels and the actual length of the same object expressed in inches. For DART files, this argument must be specified if `unitlength="mm"` or `unitlength="cm"`. For RSML files, the resolution is imported from the metadata and does not need to be specified as an argument.

`unitlength` A character string specifying the unit of length that must be used to express the results. The value acceptable for this argument could be either “px” for pixels, “cm” for centimetres or “mm” for millimetres.

`rsml.date` Only used for RSML files. A numeric value specifying the root system age (the unit of time is defined by the user). If `rsml.date=NULL`, a default value of 1 is used by the function. If the root system age is stored as a continuous function along the root segments, a character string specifying the name/label of the function can be used (see details).

`rsml.connect` Only used for RSML files. A logical value that must be specified when `is.null(inputrsml)=FALSE`. If `TRUE`, the lateral roots are connected to their parent root (see details). If `FALSE`, no connection is made between daughter and mother roots.

Details

To run `architect` efficiently, DART (.rac or .tps) and RSML (.rsml) files must have been saved with their appropriate extension.

If `inputtps` contains a single tps file, it is used by `architect` for each rac file located in `inputrac`. If `inputtps` contains more than one tps file, the number of tps files in `inputtps` must be equal to the number of rac files in `inputrac` and corresponding rac and tps files must have the same name.

After reading the RSML files located in `inputrsml`, the data associated with each root system are converted into a data frame possessing the same structure as the rac files created by DART. The `architect` function then uses these data frames to calculate RSA traits describing the global architecture of each root system.

The argument `rsml.date` is used to create a tps file for each root system located in `inputrsml`. If the root system age is stored as a continuous function along the root segments, the `rsml.date` argument can be a character string specifying the name/label of the function. For instance, the root system age at the first apparition of each node is stored as a continuous function called “age” in RSML files containing RSA data simulated with the ArchiSimple model (*Pages et al (2013)*). The “age” function attaches the root system age along the root segments, so that each node is associated with one age value. An RSML library containing 3D RSA data simulated with ArchiSimple is available here: https://doi.org/10.5281/zenodo.208214.

The `rsml.connect` argument can be used to connect the lateral roots to their corresponding mother root. If `rsml.connect=TRUE`, each point starting a lateral root is connected to the nearest point located on its mother root. DBase is approximated by the distance between the parent root base and the point located on the mother root that is closest to the point starting a lateral root. Using RSML files, only RSA data associated with roots possessing a branching order lower or equal to 5 are used for the computation of RSA traits.

Growth rates are calculated as follows:

- *First-order root growth rate*: for the first observation date, it is calculated as the ratio of the total first-order root length to the root system age. For other observation dates (t), it is calculated as the difference between the total first-order root length at time t and t-1 divided by the difference between the root system age at time t and t-1.
- *Lateral root growth rates*: for the first observation date, it is calculated as the ratio of the lateral root length to the root system age. For other observation dates (t), it is calculated as the difference between the lateral root length at time t and t-1 divided by the difference between the root system age at time t and t-1.
- *Growth rate of the root system*: for the first observation date, it is calculated as the ratio of the total root system length to the root system age. For other observation dates (t), it is calculated

as the difference between the total root system length at time t and t-1 divided by the difference between the root system age at time t and t-1.

Value

Returns a data frame summarizing all RSA traits calculated for each rac/RSML file located in `inputrac`/`inputrsml`.

The data frame contains the following columns: **FileName** (the name of the rac/RSML files), **Time** (the root system age expressed in the same unit (=unittime) as in the tps file(s) (for rac files) or in the same unit as `rsml.date` (for RSML files)), **TRL** (the total root system length expressed in `unitlength`), **GRTR** (the growth rate of the root system expressed in `unitlength`/unittime), **L1R** (the first-order root length expressed in `unitlength`), **GR1R** (the first-order root growth rate expressed in `unitlength`/unittime), **TN1R** (the total number of first-order roots), **TNLR** (the total number of lateral roots), **TLRL** (the total lateral root length expressed in `unitlength`), **NxLR** (the total number of lateral roots with a branching order equal to x), **LxLR** (the total length of lateral roots with a branching order equal to x expressed in `unitlength`), **MLxLR** (the mean length of lateral roots with a branching order equal to x expressed in `unitlength`), **GRxL** (the growth rate of lateral roots with a branching order equal to x expressed in `unitlength`/unittime), **D2LR** (the density of secondary roots on the first-order root(s) expressed in root(s)/`unitlength`).

Author(s)

Benjamin M. Delory, Caroline Baudson, Yves Brostaux, Guillaume Lobet, Loic Pages, Patrick du Jardin, Pierre Delaplace

References

Delory B.M., Baudson C., Brostaux Y., Lobet G., du Jardin P., Pages L., Delaplace P. (2016) archiDART: an R package for the automated computation of plant root architectural traits, *Plant and Soil*, DOI: 10.1007/s11104-015-2673-4.

Le Bot J., Serra V., Fabre J., Draye X., Adamowicz S., Pages L. (2010) DART: a software to analyse root system architecture and development from captured images, *Plant and Soil*, DOI: 10.1007/s11104-009-0005-2.

Lobet G., Pound M.P., Diener J., Pradal C., Draye X., Godin C., Javaux M., Leitner D., Meunier F., Nacry P., Pridmore T.P., Schnepf A. (2015) Root System Markup Language: Toward a Unified Root Architecture Description Language, *Plant Physiology*, DOI: 10.1104/pp.114.253625.

Pages L., Becel C., Boukcim H., Moreau D., Nguyen C., Voisin, A-S. (2013) Calibration and evaluation of ArchiSimple, a simple model of root system architecture, *Ecological Modelling*, DOI: 10.1016/j.ecolmodel.2013.11.014.

Examples

```
## Locate folder with DART and RSML files
path <- system.file("extdata", package="archiDART")

## Compute RSA parameters for DART files only
res2a <- architect(inputrac=path, inputtps=path, res=75, unitlength="cm")

## Compute RSA parameters for RSML files only
```

```
res2b <- architect(inputrsml=path, unitlength="cm", rsml.connect=TRUE, rsml.date="age")

## Compute RSA parameters for DART and RSML files
res2c <- architect(inputrac=path, inputtps=path, inputrsml=path, res=75, unitlength="cm",
rsml.connect=TRUE, rsml.date="age")
```

latdist *Computing Lateral Root Length and Density Distribution*

Description

Analysing the lateral root length and density distribution on each mother root using Data Analysis of Root Tracings (DART) output files and RSA data encoded with the Root System Markup Language (RSML). More information can be found in *Delory et al (2016)*, *Le Bot et al (2010)*, and *Lobet et al (2015)*.

Usage

```
latdist(inputrac=NULL, inputrsml=NULL, output=c("lrd","dtp"), res=NULL, unitlength="px",
        int.length=NULL, interpol=NULL, rsml.connect=TRUE)
```

Arguments

`inputrac` A character string specifying the path to the folder containing the rac files created by DART. This argument could be the same as `inputrsml`.

`inputrsml` A character string specifying the path to the folder containing the RSML files. This argument could be the same as `inputrac`.

`output` A character string specifying the type of output that is returned by the function. Two values are acceptable for this argument: “lrd” (default value; root length and density distribution) and “dtp” (distance between neighbouring lateral roots). See details.

`res` Mandatory for DART files only. **If images were acquired with a flatbed scanner**: a numeric value specifying the resolution of the images used to vectorize the root systems with DART (resolution is expressed in dots/inch). **If images were acquired with a digital camera**: the ratio between the length of a reference object located on the image expressed in pixels and the actual length of the same object expressed in inches. For DART files, this argument must be specified if `unitlength="mm"` or `unitlength="cm"`. For RSML files, the resolution is imported from the metadata and does not need to be specified as an argument.

`unitlength` A character string specifying the unit of length that must be used to plot the root systems. The value acceptable for this argument could be either “px” for pixels, “cm” for centimetres or “mm” for millimetres.

`int.length` Mandatory when `output="lrd"`. A numeric value specifying the interval length (expressed in `unitlength`) used to calculate local lateral root length and density (see details).

`interpol` A numeric value specifying the number of points used on each mother root to calculate local lateral root length and density. By default, `interpol=NULL` means that the function calculates local lateral root length and density at each branching point of each mother root. A numeric value can be specified when `output="lrd"` (see details).

`rsml.connect` Only used for RSML files. A logical value that must be specified when `is.null(inputrsml)=FALSE`. If `TRUE`, the lateral roots are connected to their parent root (see details). If `FALSE`, no connection is made between daughter and mother roots.

Details

DBase: distance between a branching point and the parent root base.

To run `latdist` efficiently, DART (.rac) and RSML (.rsml) files must have been saved with their appropriate extension.

After reading the RSML files located in `inputrsml`, the data associated with each root system are converted into a data frame possessing the same structure as the rac files created by DART. The `latdist` function then uses these data frames to compute lateral root length and density distribution.

The `rsml.connect` argument can be used to connect the lateral roots to their corresponding mother root. If `rsml.connect=TRUE`, each point starting a lateral root is connected to the nearest point located on its mother root. DBase is approximated by the distance between the parent root base and the point located on the mother root that is closest to the point starting a lateral root. Using RSML files, only RSA data associated with roots possessing a branching order lower or equal to 5 are used for the computation of RSA traits.

If `output="lrd"`, the function uses `interpol` to select the appropriate DBase values for which the calculation of RSA parameters should be performed. Then, a lateral root density and a total lateral root length are calculated on intervals possessing a length equal to `int.length` and centred on each DBase value.

Value

Returns a list including:

`root` A list of data frames. Each element of the list is named as its corresponding rac/RSML file and contains the following columns: **Root** (the identification number of each root constituting a vectorized root system), **Ord** (the branching order), **LatRootNum** (the number of lateral roots), **FinalRootLength** (the root length at the last observation date expressed in `unitlength`), **LatRootDensity** (the lateral root density calculated as LatRootNum/FinalRootLength and expressed in root(s)/`unitlength`).

`results` A list of secondary lists. Each element of the primary list (results) is named as its corresponding rac/RSML file. The secondary lists contain as much elements as roots constituting a vectorized root system. Each element of the secondary lists could be either a `NULL` value if a root does not have lateral roots or if no DBase value is comprised in [`int.length`/2, FinalRootLength-(`int.length`/2)], or a data frame made of two or three columns depending on the `output` value.

If `output="lrd"`: **DBase** (the distance between the branching point and the parent root base expressed in `unitlength`), **LRD** (Local lateral root density expressed in root(s)/`unitlength`), **LRL** (Local lateral root length expressed in `unitlength/unitlength`).

If `output="dtp"`: **DBase** (the distance between the branching point and the parent root base expressed in `unitlength`), **DTP** (the distance to the previous lateral root expressed in `unitlength`).

Author(s)

Benjamin M. Delory, Guillaume Lobet, Loic Pages

References

Delory B.M., Baudson C., Brostaux Y., Lobet G., du Jardin P., Pages L., Delaplace P. (2016) archiDART: an R package for the automated computation of plant root architectural traits, *Plant and Soil*, DOI: 10.1007/s11104-015-2673-4.

Le Bot J., Serra V., Fabre J., Draye X., Adamowicz S., Pages L. (2010) DART: a software to analyse root system architecture and development from captured images, *Plant and Soil*, DOI: 10.1007/s11104-009-0005-2.

Lobet G., Pound M.P., Diener J., Pradal C., Draye X., Godin C., Javaux M., Leitner D., Meunier F., Nacry P., Pridmore T.P., Schnepf A. (2015) Root System Markup Language: Toward a Unified Root Architecture Description Language, *Plant Physiology*, DOI: 10.1104/pp.114.253625.

Examples

```
## Locate folder with DART and RSML files
path <- system.file("extdata", package="archiDART")

##-----------------
## DART files only
##-----------------

## Plotting the results for the primary root
## output="lrd", interpol=NULL, int.length=5
res3a <- latdist(inputrac=path, output="lrd", res=75, unitlength="cm",
int.length=5)

plot(res3a$results$ch7[[1]]$DBase, res3a$results$ch7[[1]]$LRD, pch=16,
xlab="DBase (cm)",ylab="Lateral root density (roots/cm)", main="LRD-interpol=NULL-int.length=5",
las=1, bty="l", xaxp=c(0,90,9))

plot(res3a$results$ch7[[1]]$DBase, res3a$results$ch7[[1]]$LRL, pch=16,
xlab="DBase (cm)",ylab="Lateral root length (cm/cm)", main="LRL-interpol=NULL-int.length=5",
las=1, bty="l", xaxp=c(0,90,9))

## output="lrd", interpol=1000, int.length=5
res3b <- latdist(inputrac=path, output="lrd", res=75, unitlength="cm",
int.length=5, interpol=1000)

plot(res3b$results$ch7[[1]]$DBase, res3b$results$ch7[[1]]$LRD, pch=16,
xlab="DBase (cm)", ylab="Lateral root density (roots/cm)", main="LRD-interpol=1000-int.length=5",
las=1, bty="l", xaxp=c(0,90,9))
```

```
plot(res3b$results$ch7[[1]]$DBase, res3b$results$ch7[[1]]$LRL, pch=16,
xlab="DBase (cm)", ylab="Lateral root length (cm/cm)", main="LRL-interpol=1000-int.length=5",
las=1, bty="l", xaxp=c(0,90,9))

## output="dtp"
res3c <- latdist(inputrac=path, output="dtp", res=75, unitlength="cm")

plot(res3c$results$ch7[[1]]$DBase, res3c$results$ch7[[1]]$DTP, pch=16,
xlab="DBase (cm)", ylab="Distance to the previous root (cm)", main="DTP", las=1, bty="l",
xaxp=c(0,90,9))

##-----------------
## RSML files only
##-----------------

## output="lrd", interpol=200, int.length=1
res3d <- latdist(inputrsml=path, unitlength="cm", output="lrd", int.length=1, interpol=200,
rsml.connect=TRUE)

##output="dtp"
res3e <- latdist(inputrsml=path, output="dtp", unitlength="cm")

##---------------------
## DART and RSML files
##---------------------

## output="lrd", interpol=200, int.length=2
res3f <- latdist(inputrac=path, inputrsml=path, output="lrd", res=75, unitlength="cm",
int.length=2, interpol=200, rsml.connect=TRUE)

## output="dtp"
res3g <- latdist(inputrac=path, inputrsml=path, output="dtp", res=75, unitlength="cm")
```

trajectory *Computing Root Growth Directions and Trajectories*

Description

Calculation of root system architecture (RSA) parameters describing the directions and the trajectories of the root system growth using Data Analysis of Root Tracings (DART) output files and RSA data encoded with the Root System Markup Language (RSML). This function allows the calculation of the branching angle of each daughter root on its corresponding mother root, statistical parameters describing the curvature of each root constituting a vectorized root system, and the root tip angles relative to a vertical direction vector at each observation date. More information can be found in *Delory et al (2016)*, *Le Bot et al (2010)*, and *Lobet et al (2015)*.

Usage

```
trajectory(inputrac=NULL, inputlie=NULL, inputtps=NULL, inputrsml=NULL, res=NULL,
    unitlength="px", unitangle="d", rotation=0, l.brangle, l.curv, l.tipangle,
      rsml.date=NULL, vertical3d="y")
```

Arguments

`inputrac` A character string specifying the path to the folder containing the rac files created by DART. This argument could be the same as `inputlie/inputtps/inputrsml`.

`inputlie` A character string specifying the path to the folder containing the lie files created by DART. This argument could be the same as `inputrac/inputtps/inputrsml`.

`inputtps` A character string specifying the path to the folder containing the tps files created by DART. This argument could be the same as `inputrac/inputlie/inputrsml`.

`inputrsml` A character string specifying the path to the folder containing the RSML files. This argument could be the same as `inputrac/inputlie/inputtps`.

`res` Mandatory for DART files only. **If images were acquired with a flatbed scanner**: a numeric value specifying the resolution of the images used to vectorize the root systems with DART (resolution is expressed in dots/inch). **If images were acquired with a digital camera**: the ratio between the length of a reference object located on the image expressed in pixels and the actual length of the same object expressed in inches. For DART files, this argument must be specified if `unitlength="mm"` or `unitlength="cm"`. For RSML files, the resolution is imported from the metadata and does not need to be specified as an argument.

`unitlength` A character string specifying the unit of length that must be used to express the results. The value acceptable for this argument could be either “px” for pixels, “cm” for centimetres or “mm” for millimetres.

`unitangle` A character string specifying the unit that must be used to express the calculated angles. The value acceptable for this argument could be either “d” for degrees (default value) or “r” for radians.

`rotation` A numeric value specifying the angle (expressed in `unitangle`) that must be used to perform a clockwise rotation of 2D root systems (see details).

`l.brangle` A numeric value specifying the minimum root length used to calculate the branching angle of a daughter root on its corresponding mother root (see details).

`l.curv` A numeric value specifying the distance used to place equidistantly spaced interpolated points along each root before calculating the angles between the direction vectors of the successive links constituting a vectorized root (see details).

`l.tipangle` A numeric value specifying the distance between the root tip and an interpolated point located along a root. Each interpolated point is used by the function to calculate a direction vector following the root growth direction near the apex prior to the calculation of the root tip angle relative to a vertical direction vector at each observation date.

`rsml.date` Only used for RSML files. A numeric value specifying the root system age (the unit of time is defined by the user). If `rsml.date=NULL`, a default value of 1 is used by the function. If the root system age is stored as a continuous function along the root segments, a character string specifying the name/label of the function can be used (see details).

`vertical3d` The vertical axis for 3D RSA data (x, y, or z).

Details

To run trajectory efficiently, DART (.rac, .lie or .tps) and RSML (.rsml) files must have been saved with their appropriate extension.

If inputtps contains a single tps file, it is used by trajectory for each lie/rac file located in inputlie/inputrac. The number of lie files in inputlie must be equal to the number of rac files in inputrac and corresponding rac and lie files must have the same name. If inputps contains more than one tps file, the number of tps files in inputtps must be equal to the number of rac/lie files in inputrac/inputlie and corresponding rac/lie and tps files must have the same name.

After reading the RSML files located in inputrsml, the data associated with each root system are converted into data frames possessing the same structure as the lie/rac files created by DART. The trajectory function then uses these data frames to compute RSA traits.

The argument rsml.date is used to create a tps file for each root system located in inputrsml. If the root system age is stored as a continuous function along the root segments, the rsml.date argument can be a character string specifying the name/label of the function. For instance, the root system age at the first apparition of each node is stored as a continuous function called “age” in RSML files containing RSA data simulated with the ArchiSimple model (*Pages et al (2013)*). The “age” function attaches the root system age along the root segments, so that each node is associated with one age value. An RSML library containing 3D RSA data simulated with ArchiSimple is available here: https://doi.org/10.5281/zenodo.208214.

The rsml.connect argument can be used to connect the lateral roots to their corresponding mother root. If rsml.connect=TRUE, each point starting a lateral root is connected to the nearest point located on its mother root. DBase is approximated by the distance between the parent root base and the point located on the mother root that is closest to the point starting a lateral root. Using RSML files, only RSA data associated with roots possessing a branching order lower or equal to 5 are used for the computation of RSA traits.

Due to technical limitations, it is sometimes easier to take a picture of a root system if it is rotated before image acquisition. In this case, the vectorized root system depicts a rotation compared with the natural plant root system. In order to make a correction, one can use rotation to specify an angle value expressed in unitangle to rotate the vectorized root system clockwise before plotting.

Hereafter, we will consider that the normal vector that is orthogonal to a direction vector (a,b) has the following coordinates: (b,-a). The direction of lateral root growth is only computed for 2D root systems. A daughter root is considered to grow at the left side of the mother root if the scalar product between a direction vector (=vector A) going from the branching point (X0) to the following point (Xd) on the daughter root and a vector (=vector B) normal to a direction vector (=vector C) going from the branching point (X0) to the following point on the mother root (Xm) is positive. If the scalar product between A and B is negative, the daughter root is considered to grow at the right side of the mother root. If the scalar product between A and B is equal to zero, the calculation of the scalar product between A and B will be performed again using a vector A going from the branching point (X0) to a point located on the daughter root at Xd<-Xd+1 until the calculated scalar product is different from zero. If A is always normal to B, a random lateral root growth orientation (either left or right) is defined. See table 2 and figure 1 of *Delory et al (2016)* for more information.

The branching angle of a daughter root on its mother root is approximated by the angle between two direction vectors going from the branching point to an interpolated point located at a distance l.brangle from the branching point on the mother or on the daughter root, respectively. The distance l.brangle between a branching point and an interpolated point is measured along the mother and the daughter roots. A first-order root having no mother root, the reported angle is the angle measured between a vertical direction vector and a second direction vector going from the first node of the root to an interpolated point located at a distance `l.brangle` from the first node on the first-order root. If a root has a final length lower than `l.brangle`, no branching angle can be calculated and the function returns a `NA` value for Branching.Angle. See table 2 and figure 1 of *Delory et al (2016)* for more information.

The curvature of each root is evaluated by the mean and the standard deviation of the local angles formed by the direction vectors of the successive links constituting a vectorized root. First, the function uses `l.curv` to interpolate equidistantly spaced points along each root constituting a vectorized root system. Second, the angles between the direction vectors of successive links along each root are calculated. Then, the mean and the standard deviation of the calculated angles are determined for each root. If a root has a final length lower than 2*`l.curv`, the function returns a `NA` value for Mean.Curv and SD.Curv. If a root has a final length lower than 3*`l.curv`, the function returns a `NA` value for SD.Curv as no standard deviation can be calculated on a single angle value. See table 2 and figure 1 of *Delory et al (2016)* for more information.

Value

Returns a list including:

`root` A list of data frames. Each element of the list is named as its corresponding rac/lie/RSML file and contains the following columns: **Root** (the identification number of each root constituting the vectorized root system), **Mother** (the parent root identification number), **Ord** (the branching order), **DBase** (the distance between the branching point to the parent root base expressed in `unitlength`), **FinalRootLength** (the root length at the last observation date expressed in `unitlength`), **Tortuosity** (the ratio of the final root length to the Euclidean distance between the branching point and the apex of the root), **Orientation** (the direction of the lateral root growth; only for 2D root systems), **Branching.Angle** (the branching angle expressed in `unitangle`), **Mean.Curv** (the mean of the local angle values calculated between the direction vectors of the successive links constructed using equidistantly spaced interpolated points along each root of a vectorized root system; the calculated mean is expressed in `unitangle`), **SD.Curv** (the standard deviation of the local angle values calculated between the direction vectors of the successive links constructed using equidistantly spaced interpolated points along each root of a vectorized root system; the calculated standard deviation is expressed in `unitangle`).

`tip` A list of data frames. Each element of the list is named as its corresponding rac/lie/RSML file and contains the following columns: **Root** (the identification number of each root constituting the vectorized root system), **Ang.DateX** (the calculated root tip angle relative to a vertical direction vector at the observation date X expressed in `unitangle`).

Author(s)

Benjamin M. Delory, Guillaume Lobet, Loic Pages

References

Delory B.M., Baudson C., Brostaux Y., Lobet G., du Jardin P., Pages L., Delaplace P. (2016) archiDART: an R package for the automated computation of plant root architectural traits, *Plant and Soil*, DOI: 10.1007/s11104-015-2673-4.

Le Bot J., Serra V., Fabre J., Draye X., Adamowicz S., Pages L. (2010) DART: a software to analyse root system architecture and development from captured images, *Plant and Soil*, DOI: 10.1007/s11104-009-0005-2.

Lobet G., Pound M.P., Diener J., Pradal C., Draye X., Godin C., Javaux M., Leitner D., Meunier F., Nacry P., Pridmore T.P., Schnepf A. (2015) Root System Markup Language: Toward a Unified Root Architecture Description Language, *Plant Physiology*, DOI: 10.1104/pp.114.253625.

Pages L., Becel C., Boukcim H., Moreau D., Nguyen C., Voisin, A-S. (2013) Calibration and evaluation of ArchiSimple, a simple model of root system architecture, *Ecological Modelling*, DOI: 10.1016/j.ecolmodel.2013.11.014.

Examples

```
## Locate folder with DART and RSML files
path <- system.file("extdata", package="archiDART")

##-----------------
## DART files only
##-----------------

res4a <- trajectory(inputrac=path, inputlie=path, inputtps=path, res=75,
unitlength="cm", l.brangle=1, l.curv=1, l.tipangle=0.5)

## Distribution of the calculated branching angles
hist(res4a$root$ch7$Branching.Angle, breaks=seq(from=0, to=180, by=5),
main="ch7-Branching angle distribution", xlab="Branching angle (d)", las=1,
xaxp=c(0,150,15), xlim=c(0,150))

## Evolution of the root tip angle for the primary root (black) and the two longest lateral roots
## (green and red)
date<-c(1:31)
plot(date, res4a$tip$ch7[1,2:ncol(res4a$tip$ch7)], type="l", lwd=2, bty="l",
las=1, ylim=c(0,90), ylab="ch7 - Root tip angle (d)", xlab="Time (Num)", col="black",
main="Root tip angle", yaxp=c(0,90,9))
lines(date, res4a$tip$ch7[206,2:ncol(res4a$tip$ch7)], lwd=2, col="green")
lines(date, res4a$tip$ch7[221,2:ncol(res4a$tip$ch7)], lwd=2, col="red")

##---------------------
## DART and RSML files
##---------------------

res4b <- trajectory(inputrac=path, inputlie=path, inputtps=path, inputrsml=path, res=75,
unitlength="cm", rsml.date="age", l.brangle=1, l.curv=1, l.tipangle=0.5, vertical3d="y")

## Branching angles of lateral roots
monocot<-res4b$root$'monocot-archisimple'
data<-data.frame(Factor=c(rep("Monocot", nrow(monocot))), monocot)
```

```
boxplot(Branching.Angle[Ord==2]~Factor[Ord==2], data=data, ylab="Branching angle (d)",
ylim=c(45,70), main="Branching angle laterals - Monocot", las=1, outline=FALSE,
range=0, notch=TRUE)

## Evolution of the root tip angle for the first-order roots of the monocot root system
tip<-res4b$tip$'monocot-archisimple'
root<-res4b$root$'monocot-archisimple'
date<-c(1:(ncol(tip)-1))
firstorderroots<-which(root$Ord==1)
colors<-colorRampPalette(c("red", "green"))(length(firstorderroots))

plot(date, tip[1,2:ncol(tip)], type="n", lwd=2, bty="l", las=1, ylim=c(0,90),
yaxp=c(0,90,9), xaxp=c(0,16,16), ylab="Root tip angle (d)", xlab="Time (Num)", col="black",
main="Monocot - Root tip angle first-order roots")

for (i in 1:length(firstorderroots)){lines(date, tip[firstorderroots[i], 2:ncol(tip)], lwd=2,
col=colors[i])}
```

Index