Package ‘PROTOLIDAR’

February 19, 2015

Type Package
Title PRocess TOol LIdar DAta in R
Version 0.1
Date 2012-11-21
Author Monica Fernanda Rinaldi
Maintainer Monica Fernanda Rinaldi <monica.rinaldi@gmail.com>
Description PROTOLIDAR package contains functions for analyze the LIDAR scan of plants (grapevine) and make 3D maps in GRASS GIS.
License GPL
LazyLoad yes
Repository CRAN
Date/Publication 2013-01-09 19:35:43
NeedsCompilation no

R topics documented:

PROTOLIDAR-package .................................................. 2
Extract_plant_3D_function ............................................ 4
Extract_plant_grapevine_function ................................. 5
Height_canopy_function .............................................. 7
LAI_function ............................................................ 8
LIDAR_data .............................................................. 9
LWA_lidar_function .................................................... 11
Number_lidar_points_into_canopy_function ...................... 12
Replicate_plants_function ........................................... 13
Rotate_function ....................................................... 14
TRV_lidar_function ................................................... 16
Width_canopy_function ............................................... 17

Index 19
Description

 PROTOLIDAR package contains functions for analyze the LIDAR scan of plants (grapevine) and make with the outputs 3D maps in GRASS-GIS.

Details

This package help to analyze the LIDAR scan and extract the grapevine plant for see the plant in 3D GRASS GIS maps.

The package contains the following dataset and functions:

LIDAR_data is the dataset of the LIDAR scan. Represent the grapevine plant (BBCH 65).

Extract_plant_grapevine_function: which cuts the excess data.

Height_canopy_function: to measure the height of canopy from the LIDAR scan.

Width_canopy_function: to measure the width of canopy from the LIDAR scan.

Number_LIDAR_points_function: to calculate the number of points into the canopy.

LAI_function: to calculate the leaf area index.

LWA_lidar_function: to calculate the leaf wall area.

TRV_lidar_function: to calculate tree row volume in m^3*ha^-1.

Rotate_function: to rotate plants to match with the planting line.

Replicate_plants_function: to replicate plants.

Author(s)

Monica Fernanda Rinaldi<monica.rinaldi@gmail.com>
Emilio Gil<emilio.gil@upc.edu>
Jordi Llorens<jordi.llorens.calveras@upc.edu>

Maintainer: Monica Fernanda Rinaldi<monica.rinaldi@gmail.com>
References


See Also

PROTOLIDAR-package

Examples

```r
## Should be DIRECTLY executable !!
## For example:
data(LIDAR_data)
x <- LIDAR_data[,1]
y <- LIDAR_data[,2]
z <- LIDAR_data[,3]
zdistance <- 190 # total LIDAR scan distance measured in cm.
miny <- 0 # minimum height of the plant measured in cm.
maxy <- 2000 # maximum height of the plant measured in cm.
minx <- 450 # minimum width from where LIDAR starts to measure (cm).
maxx <- 1470 # maximum width from where LIDAR starts to measure (cm).
minz <- 0 # the beginning of the LIDAR scan measured in cm.
maxz <- 186 # the end of the LIDAR scan measured in cm (length of interest).

## The function is currently defined as
Extract_plant_grapevine_function <- function(x,y,z,zdistance,miny,maxy,minx,maxx,minz,maxz){
    y <- -y
    y <- y-min(y)
z <- (z*zdistance)/max(z)
x_cm <- 0
y_cm <- 0
z_cm <- 0
for (i in 1:length(x)){
    if (x[i] >= minx && x[i] <= maxx && y[i] >= miny && y[i] <= maxy && z[i] >= minz && z[i] <= maxz) {
        y_cm[i] <- y[i]
x_cm[i] <- x[i]
z_cm[i] <- z[i]
    }
}
y_cm <- na.omit(y_cm[2:length(y_cm)])
y_cm <- as.numeric((y_cm-min(y_cm))/1000)
x_cm <- as.numeric(na.omit(x_cm[2:length(x_cm)])/1000)
z_cm <- as.numeric(na.omit(z_cm[2:length(z_cm)])/1000)
return <- data.frame(x_cm,y_cm,z_cm)
}
out <- Extract_plant_grapevine_function(x,y,z,zdistance,miny,maxy,minx,maxx,minz,maxz)
x = out[,1]
y = out[,2]
z = out[,3]
```
Extract_plant_3D_function

Extract plant 3D (grapevine).

Description
This function move the axes x,y,z to the center of the plant. This output could be exported and transformed in GRASS GIS in 3D maps.

Usage
Extract_plant_3D_function(out, z_min, z_max, y_min, y_max, distance_left, distance_right)

Arguments
- `out` out is a data frame output from Extract_plant_grapevine_function.
- `z_min` the minimum position of the stem in z measured in meters.
- `z_max` the maximum position of the stem in z measured in meters.
- `y_min` the minimum position of the stem in y measured in meters.
- `y_max` the maximum position of the stem in y measured in meters.
- `distance_left` the left distance of the plant. generally here we can write the half of the distance between plants, but is better the real distance from the center of the plant. Measured in meters.
- `distance_right` the right distance of the plant, generally here we can write the half of the distance between plants, but is better the real distance from the center of the plant. Measured in meters.

Author(s)
Monica Fernanda Rinaldi

Examples

```r
# Should be DIRECTLY executable !!
## out come from Extract_plant_grapevine_function. The other parameters or inputs are needed to write before.
## For example:
data (LIDAR_data)
x <- LIDAR_data[,1]
y <- LIDAR_data[,2]
z <- LIDAR_data[,3]
zdistance <- 190 # total LIDAR scan distance measured in cm.
```
Extract_plant_grapevine_function

Extract vine plant from the entire dataset.

Description

The function cut the plant at fixes values of x, y and z. Where x is width, y is height and z is front view or path of the tractor.
Usage

```
Extract_plant_grapevine_function(x, y, z, zdistance, miny, maxy, minx, maxx, minz, maxz)
```

Arguments

- **x**: the width of the plant measured with LIDAR scan in cm.
- **y**: the height of the plant measured with LIDAR scan in cm.
- **z**: the front of the plant or path of the tractor measured with LIDAR scan in cm.
- **zdistance**: the z distance of LIDAR scan measured in cm.
- **miny**: the minimum height at which we cut the plant measured in cm.
- **maxy**: the maximum height at which we cut the plant measured in cm.
- **minx**: the minimum width to which we want to measure the plant measured in cm.
- **maxx**: the maximum width to which we want to measure the plant measured in cm.
- **minz**: the minimum distance at which we cut the plant, measured in cm.
- **maxz**: the maximum distance at which we cut the plant, measured in cm.

Details

Path or direction of the tractor at constant velocity.

Author(s)

Monica Fernanda Rinaldi

Examples

```
## Should be DIRECTLY executable !! --
## First needed the LIDAR_data scan (that is one dataframe with x,y,z columns).
## Second needed define these inputs in cm: zdistance,miny,maxy,minx,maxx,minz,maxz.
## For example:
data (LIDAR_data)
x <- LIDAR_data [,1]
y <- LIDAR_data [,2]
z <- LIDAR_data [,3]
zdistance <- 190 # total LIDAR scan distance measured in cm.
miny <- 0 # minimum height of the plant measured in cm.
maxy <- 2000 # maximum height of the plant measured in cm.
minx <- 450 # minimum width from where LIDAR starts to measure (cm).
maxx <- 1470# maximum width from where LIDAR starts to measure (cm).
minz <- 0 # the beginning of the LIDAR scan measured in cm.
maxz <- 186 # the end of the LIDAR scan measured in cm (length of interest).

## The function is currently defined as
Extract_plant_grapevine_function <- function(x,y,z,zdistance,miny,maxy,minx,maxx,minz,maxz){
  y <- -y
  y <- y-min(y)
  z<- (z*zdistance)/max(z)
  x_cm <- 0
```
**Height_canopy_function**

*Height of the canopy measured with LIDAR scan.*

**Description**

From the LIDAR dataset can be calculate the height of the grapevine plant. The function returns the average, minimum and maximum value of the height measured in meters.

**Usage**

```r
height_canopy_function(data_3D, distance_left, distance_right, min_canopy, max_canopy)
```

**Arguments**

- **data_3D**: data_3D is the output from `Extract_plant_3D_function`.
- **distance_left**: the left distance of the plant, generally here we can write the half of the distance between plants, but is better the real distance from the center of the plant. Measured in meters.
- **distance_right**: the right distance of the plant, generally here we can write the half of the distance between plants, but is better the real distance from the center of the plant. Measured in meters.
- **min_canopy**: the minimum height of the canopy, measured in meters.
- **max_canopy**: the maximum height of the canopy, measured in meters.
Details

Maximum and minimum values of height of canopy could be approximative values.

Author(s)

Monica Fernanda Rinaldi

Examples

```r
# Should be DIRECTLY executable !! ----
# Data_3D is the output from the Extrac_plant_3D_function.
# For example:
data (LIDAR_data)
x <- LIDAR_data[,1]
y <- LIDAR_data[,2]
z <- LIDAR_data[,3]
zdistance <- 190 # total LIDAR scan distance measured in cm.
miny <- 0 # minimum height of the plant measured in cm.
maxy <- 2000 # maximum height of the plant measured in cm.
minx <- 450 # minimum width from where LIDAR starts to measure (cm).
maxx <- 1470# maximum width from where LIDAR starts to measure (cm).
minz <- 0 # the beginning of the LIDAR scan measured in cm.
maxz <- 186 # the end of the LIDAR scan measured in cm (length of interest).
out <- Extract_plant_grapevine_function(x,y,z,zdistance,miny,maxy,minx,maxx,minz,maxz)
z_min <- 1.1
z_max <- 1.13
y_min <- 0.4
y_max <- 0.5
distance_left <- -0.6
distance_right <- 0.51
data_3D <- Extract_plant_3D_function(out,z_min,z_max,y_min,y_max,distance_left,distance_right)
min_canopy <- 0.4 # is the minimum height of the canopy, approximately. Measured in meters.
max_canopy <- 2  # is the maximum height of the canopy, approximately. Measured in meters.

# The function is currently defined as
Height_canopy_function <- function(data_3D,distance_left,distance_right,min_canopy,max_canopy){
x_plant <- y_plant <- z_plant <- NULL
canopy <- subset(data_3D, data_3D$z_plant >= distance_left & data_3D$z_plant <= distance_right & data_3D$y_plant <= max_canopy)
mean_height_canopy <- mean(canopy[,2])
min_height_canopy <- min(canopy[,2])
max_height_canopy <- max(canopy[,2])
return(data.frame(mean_height_canopy,min_height_canopy,max_height_canopy))
}
height_canopy <- Height_canopy_function(data_3D,distance_left,distance_right,min_canopy,max_canopy)[,1]
```

LAI_function

Leaf Area Index (LAI) calculate from LIDAR scan.
LIDAR data

Description

The LAI need in inputs the number of leaves, the leaf area (m^2), row distance (m) and in row spacing (m).

Usage

LAI_function(Number_of_leaves_by_plant, Leaf_Area, row_distance, in_row_distance)

Arguments

Number_of_leaves_by_plant
  here need count the number of leaves of the plant.
Leaf_Area
  here need calculate the leaf area, measured in m^2.
row_distance
  the row distance of the orchard measured in meters.
in_row_distance
  the in row distance or distance between plants of the orchard measured in meters.

Author(s)

Monica Fernanda Rinaldi

Examples

```r
## Should be DIRECTLY executable !! ----
## Here needed some inputs measured manually like leaf area (m^2) and number of leaves.
number_of_leaves <- 420
leaf_area <- 0.010 ## measured in m^2.
row_distance <- 2.9 ## measured in meters.
in_row_distance <- 1.4 ## measured in meters.

## The function is currently defined as
LAI_function <- function(Number_of_leaves_by_plant, Leaf_Area, in_row_distance){
  LAI <- Number_of_leaves_by_plant * Leaf_Area / in_row_distance
  return(LAI)
}
LAI_function(number_of_leaves, leaf_area, in_row_distance)
```

LIDAR data

Description

LIDAR scan dataset in BBCH 65 (grapevine). Where x is width, y is height and z is front view or path of the tractor.
Usage

data(LIDAR_data)

Format

A data frame with 10108 observations on the following 3 variables.

v1  a numeric vector that represents x value or width
v2  a numeric vector that represents y value or height
v3  a numeric vector that represents z value or front view

Details

The laser scanner used was a LMS-200 model (Sick, Dusseldorf, Germany), a fully-automatic divergent laser scanner based on the measurement of time-of-flight (TOF) with an accuracy of 15 mm in a single shoot measurement and 5 mm standard deviation in a range up to 8m. The time between the transmission and the reception of the pulsed near-infrared laser beam is used to measure the distance between the scanner and the reflecting object surface. The laser beam is deflected by a rotating mirror turning at 4500 rpm, which results in a fan shaped scan pattern where the maximum scanning angle is 180 degree.

Source

LIDAR scan in BBCH 65 stage.

References


Examples

```r
## LIDAR_data is the input to Extract_plant_grapevine_function.

data(LIDAR_data, package = 'PROTOLIDAR')

x = LIDAR_data[,1]
y = LIDAR_data[,2]
z = LIDAR_data[,3]
```
LWA_lidar_function  

Leaf Wall Area (LWA) measured in $m^2*ha^{-1}$.

Description

LWA need as inputs the height of canopy (m) the ground area (generally one hectare, measured in $m^2$) and the row spacing (m).

Usage

LWA_lidar_function(height_canopy, ground_area, row_spacing)

Arguments

- height_canopy: height of canopy manually or measured with Height_canopy_function.
- ground_area: is the orchard area measured in $m^2$, generally one hectare.
- row_spacing: row spacing measured in meters.

Author(s)

Monica Fernanda Rinaldi

References


Examples

```r
## Should be DIRECTLY executable !! ----
height_canopy = 2  ## this value is the maximum of Height_canopy_function.
ground_area = 10000  ## generally is one hectare in m^2.
row_spacing = 2.9  ## measured in meters.

## The function is currently defined as
function(height_canopy,ground_area,row_spacing){

LWA <- 2* height_canopy * (ground_area/row_spacing)

return(LWA)
}

LWA_lidar_function(height_canopy,ground_area,row_spacing)
```
Number_lidar_points_into_canopy_function

Number of LIDAR points into the canopy.

Description

This function describe the number of points measured with LIDAR scan into the canopy.

Usage

Number_lidar_points_into_canopy_function(data_3D, distance_left, distance_right, min_canopy, max_canopy)

Arguments

data_3D: data_3D is the output from Extract_plant_grapevine_function.
distance_left: the left distance of the plant, generally here we can write the half of the distance between plants, but is better the real distance from the center of the plant. Measured in meters.
distance_right: the right distance of the plant, generally here we can write the half of the distance between plants, but is better the real distance from the center of the plant. Measured in meters.
min_canopy: the minimum height of the canopy.
max_canopy: the maximum height of the canopy.

Author(s)

Monica Fernanda Rinaldi

Examples

```r
# Should be DIRECTLY executable !! ----
# For example:
data (LIDAR_data)
x <- LIDAR_data[,1]
y <- LIDAR_data[,2]
z <- LIDAR_data[,3]
zdistance <- 190 # total LIDAR scan distance measured in cm.
miny <- 0 # minimum height of the plant measured in cm.
maxy <- 2000 # maximum height of the plant measured in cm.
minx <- 450 # minimum width from where LIDAR starts to measure (cm).
maxx <- 1470# maximum width from where LIDAR starts to measure (cm).
minz <- 0 # the beginning of the LIDAR scan measured in cm.
maxz <- 186 # the end of the LIDAR scan measured in cm (length of interest).
out <- Extract_plant_grapevine_function(x,y,z,zdistance,miny,maxy,minx,maxx,minz,maxz)
z_min <- 1.1
z_max <- 1.13
y_min <- 0.4
```
Replicate_plants_function

Replicate plants function.

Description

This function helped to make 3D maps in GRASS GIS when you have only one scan of a plant. First needed rotate the plants and then could be replicate each plant in the row.

Usage

Replicate_plants_function(plants_rotate, data, latitude, longitude)

Arguments

- plants_rotate: here need use the output of Rotate_function.
- data: here need use the output of Extract_plant_3D_function and Extract_plant_grapevine_function.
- latitude: here need the latitudine of each plant.
- longitude: here need the longitude of each plant.

Author(s)

Monica Fernanda Rinaldi

Examples

## Should be DIRECTLY executable !!
## out come from Extract_plant_grapevine_function. The other parameters or inputs are needed to write before.
## For example:
  data (LIDAR_data)
  x <- LIDAR_data[,1]
  y <- LIDAR_data[,2]
Rotate function

The function help in rotate the plants to match with the planting line.
Rotate_function

Usage

Rotate_function(data_3D, angle)

Arguments

data_3D  data_3D is the output of Extract_plant_3D_function.
angle    angle is one value like 14.96 degree that needed rotate the plants.

Author(s)

Monica Fernanda Rinaldi

Examples

## Should be DIRECTLY executable !!
## out come from Extract_plant_grapevine_function. The other parameters or inputs are needed to write before.
## For example:

data (LIDAR_data)
x <- LIDAR_data[,1]
y <- LIDAR_data[,2]
z <- LIDAR_data[,3]
zdistance <- 190 # total LIDAR scan distance measured in cm.
miny <- 0 # minimum height of the plant measured in cm.
maxy <- 2000 # maximum height of the plant measured in cm.
minx <- 450 # minimum width from where LIDAR starts to measure (cm).
maxx <- 1470 # maximum width from where LIDAR starts to measure (cm).
minz <- 0 # the beginning of the LIDAR scan measured in cm.
maxz <- 186 # the end of the LIDAR scan measured in cm (length of interest).
out <- Extract_plant_grapevine_function(x,y,z,zdistance,miny,maxy,minx,maxx,minz,maxz)
z_min <- 1.1
z_max <- 1.13
y_min <- 0.4
y_max <- 0.5
distance_left <- -0.6
distance_right <- 0.51
data_3D <- Extract_plant_3D_function(out,z_min,z_max,y_min,y_max,distance_left,distance_right)
angle <- 14.96

## The function is currently defined as
Rotate_function <- function(data_3D,angle){
  z <- -data_3D[,3]
x <- data_3D[,1]
x_rot <-c(x*cos(angle)-z*sin(angle))
y_rot <-c(x*sin(angle)+z*cos(angle))
return(data.frame(x_rot,y_rot))
}
Plants_rotate <- Rotate_function(data_3D,angle)
x_rot <- Plants_rotate[,1]
y_rot <- Plants_rotate[,2]
##plot
plot(x_rot,y_rot)
Tree Row Volume (TRV).

**Description**

TRV measured in m^3*ha^-1.

**Usage**

\[ TRV_{lidar\_function}(height\_canopy, width\_canopy, row\_spacing) \]

**Arguments**

- `height\_canopy` use `Height\_canopy\_function`, measured in meters.
- `width\_canopy` use `Width\_canopy\_function`, measured in meters.
- `row\_spacing` row spacing measured in meters.

**Author(s)**

Monica Fernanda Rinaldi

**References**


**Examples**

```r
## Should be DIRECTLY executable !! ----
## Here need use: Height\_canopy\_function and Width\_canopy\_function or values measured manually.
height\_canopy <- 1.995 ## the value is the result of Height\_canopy\_function.
width\_canopy <- 0.426 ## the value is the result of Width\_canopy\_function.
row\_spacing = 2.9 ## measured in meters.

## The function is currently defined as
TRV\_lidar\_function <- function(height\_canopy, width\_canopy, row\_spacing){
  TRV <- height\_canopy * width\_canopy * 10000 / row\_spacing
  return(TRV)
}
TRV <- TRV\_lidar\_function(height\_canopy, width\_canopy, row\_spacing)
```
Width_canopy_function  Canopy width measured with LIDAR.

Description
From the LIDAR dataset can be calculate the width of the grapevine plant. The function returns the average, minimum and maximum value of the width measured in meters.

Usage
Width_canopy_function(data_3D, distance_left, distance_right, min_canopy, max_canopy)

Arguments
- data_3D: Here need use the output of the Extract_plant_3D_function.
- distance_left: the left distance of the plant, generally here we can write the half of the distance between plants, but is better the real distance from the center of the plant. Measured in meters.
- distance_right: the right distance of the plant, generally here we can write the half of the distance between plants, but is better the real distance from the center of the plant. Measured in meters.
- min_canopy: the minimum height of the canopy, measured in meters.
- max_canopy: the maximum height of the canopy, measured in meters.

Details
Maximum and minimum values of height of canopy could be approximative values.

Author(s)
Monica Fernanda Rinaldi

Examples
## Should be DIRECTLY executable !! ----
## Data_3D is the output from the Extract_plant_3D_function.
## For example:
data (LIDAR_data)
x <- LIDAR_data[,1]
y <- LIDAR_data[,2]
z <- LIDAR_data[,3]
zdistance <- 190 # total LIDAR scan distance measured in cm.
miny <- 0 # minimum height of the plant measured in cm.
maxy <- 2000 # maximum height of the plant measured in cm.
minx <- 450 # minimum width from where LIDAR starts to measure (cm).
maxx <- 1470 # maximum width from where LIDAR starts to measure (cm).
minz <- 0 # the beginning of the LIDAR scan measured in cm.
maxz <- 186 # the end of the LIDAR scan measured in cm (length of interest).
out <- Extract_plant_grapevine_function(x,y,z,z_distance,miny,maxy,minx,maxx,minz,maxz)
z_min <- 1.1
z_max <- 1.13
y_min <- 0.4
y_max <- 0.5
distance_left <- -0.6
distance_right <- 0.51
data_3D <- Extract_plant_3D_function(out,z_min,z_max,y_min,y_max,distance_left,distance_right)
min_canopy <- 0.4 # is the minimum height of the canopy, approximately. Measured in meters.
max_canopy <- 2 # is the maximum height of the canopy, approximately. Measured in meters.

## The function is currently defined as
Width_canopy_function <- function(data_3D,distance_left,distance_right,min_canopy,max_canopy){
  x_plant <- y_plant <- z_plant <- NULL
  canopy <- subset(data_3D, data_3D$z_plant >= distance_left & data_3D$z_plant <= distance_right & data_3D$y_plant <= -0.6 & data_3D$y_plant >= 0.51)
  mean_width_canopy <- mean(abs(canopy[,1]))
  min_width_canopy <- min(abs(canopy[,1]))
  max_width_canopy <- max(abs(canopy[,1]))
  return(data.frame(mean_width_canopy, min_width_canopy, max_width_canopy))
}

width_canopy <- Width_canopy_function(data_3D,distance_left,distance_right,min_canopy,max_canopy)[,1]
Index

*Topic **aplot**
  Extract_plant_3D_function, 4
  Extract_plant_grapevine_function, 5
  PROTOLIDAR-package, 2
  Replicate_plants_function, 13
  Width_canopy_function, 17

*Topic **array**
  Extract_plant_3D_function, 4
  Extract_plant_grapevine_function, 5
  Height_canopy_function, 7
  LAI_function, 8
  PROTOLIDAR-package, 2
  Replicate_plants_function, 13
  Rotate_function, 14
  TRV_lidar_function, 16
  Width_canopy_function, 17

*Topic **datasets**
  LIDAR_data, 9
  PROTOLIDAR-package, 2

*Topic **math**
  Height_canopy_function, 7
  LWA_lidar_function, 11
  Number_lidar_points_into_canopy_function, 12
  PROTOLIDAR-package, 2
  Rotate_function, 14

  Extract_plant_3D_function, 4
  Extract_plant_grapevine_function, 5

  Height_canopy_function, 7

  LAI_function, 8
  LIDAR_data, 9
  LWA_lidar_function, 11

  Number_lidar_points_into_canopy_function, 12

  PROTOLIDAR (PROTOLIDAR-package), 2
  Replicate_plants_function, 13
  Rotate_function, 14
  TRV_lidar_function, 16
  Width_canopy_function, 17

19