Package 'waveformlidar'

August 1, 2020

Type Package

Title Waveform LiDAR Data Processing and Analysis

Version 1.2.0

Date 2020-07-25

Depends R (>= 4.0.0)

Maintainer Tan Zhou <tankchow12@gmail.com>

Imports data.table, caTools, minpack.lm, flux, raster, rgdal, sp, rgeos, sqldf, splitstackshape, reshape2

Suggests knitr, rPeaks

Additional_repositories https://tankwin08.github.io/drat

Description

A wealth of Full Waveform (FW) Light Detection and Ranging (LiDAR) data are available to the public from different sources, which is poised to boost the extensive application of FW LiDAR data. However, we lack a handy and open source tool that can be used by ecological and remote sensing communities for processing and analyzing FW LiDAR data. To this end, we introduce 'waveformlidar', an R package dedicated to FW LiDAR processing, analysis and visualization as a solution to the constraint. Specifically, this package provides several commonly used waveform processing methods such as Gaussian, adaptive Gaussian and Weibull decompositions, and deconvolution approaches (Gold and Richard-Lucy (RL)) with customized settings. In addition, we also develop some functions to derive commonly used waveform metrics for characterizing vegetation structure. Moreover, a new way to directly visualize FW LiDAR data is developed through converting waveforms into points to form the Hyper Point cloud (HPC), which can be easily adopted and subsequently analyzed with existing discrete-return LiDAR processing tools such as 'LAStools' and 'FUSION'. Basic explorations of the HPC such as 3D voxelization of the HPC and conversion from original waveforms to composite waveforms are also available in this package. All of these functions are developed based on small-footprint FW LiDAR data, but they can be easily transplanted to the large footprint FW LiDAR data such as Geoscience Laser Altimeter System (GLAS) and Global Ecosystem Dynamics Investigation (GEDI) data analysis. References: Zhou et al. (2017a) <doi:10.1016/j.isprsjprs.2017.04.021>; Zhou et al. (2017b) <doi:10.1016/j.rse.2017.08.012>; Zhou et al. (2018a) <doi:10.3390/rs10010039>;Zhou et al. (2018b) <d

License GPL (>= 2)

Encoding UTF-8

LazyData true

URL https://github.com/tankwin08/waveformlidar,

BugReports https://github.com/tankwin08/waveformlidar/issues

RoxygenNote 7.1.0

NeedsCompilation no

Author Tan Zhou [aut, cre]

Repository CRAN

Date/Publication 2020-08-01 09:20:03 UTC

R topics documented:

---

agennls *agennls*

Description

The function allows you to prepare the formula and start values for adaptive or generalized Gaussian decomposition using nlsLM.

Usage

```
agennls(A, u, sig, r)
```

Arguments

A	is the amplitude of waveform.
u	is the peak location of waveform.
sig	is the echo width of waveform.
r	is the rate parameter of adpative or generalized Gaussian distribution. For Gaussian distributionm, r is fixed to 2.

Value

return a formula suitable for different number of waveform componments with adaptive Gaussian distribution.

Examples

```
###these four should have the same length
A<-c(76,56,80);u<-c(29,40,67);sig<-c(4,3,2.5); r<-c(2,2,2)
fg<-agennls(A,u,sig,r)

##input formula for Gaussian decomposition
fgf<-fg$formula
###start values
fgs<-fg$start
```

decom *decom*

Description

The function allows you to eatimate parameters charcterizing waveforms and to pave the way for generating waveform-based point cloud.

Usage

```
decom(x, smooth = TRUE, width = 3, thres = 0.22)
```

Arguments

x	is a waveform with a index at the begining and followed with intensities.
smooth	is tell whether you want to smooth the waveform to remove some obvious outliers. Default is TRUE.
width	width of moving window.Default is 3, must be odd integer between 1 and n.This parameter ONLY work when the smooth is TRUE.
thres	is to determine if the detected peak is the real peak whose intensity should be higher than threshold*maximum intensity. Default is 0.22.

Value

A list contains estimates of A, u, sig after decomposition.

References

Zhou, Tan*, Sorin C. Popescu, Keith Krause, Ryan D. Sheridan, and Eric Putman, 2017. Gold-A novel deconvolution algorithm with optimization for waveform LiDAR processing. ISPRS Journal of Photogrammetry and Remote Sensing 129 (2017): 131-150. https://doi.org/10.1016/j.isprsjprs.2017.04.021

Examples

```
##import return waveform data
library(data.table)
data(return)
return<-data.table(index=c(1:nrow(return)),return)
x<-return[1,] ###must be a dataset including intensity with index at the beginning.
r1<-decom(x)
r2<-decom(x,smooth=TRUE,width=5) ###you can assign different smooth width for the data
###when it comes very noisy waveforms, it may give you some problems
xx<-return[182,]
r3<-decom(xx)  ##this one returns NULL which means the function didn't work for the
               ##complex waveform or too noisy waveform,we should try to reprocess
               ##these unsucessful waveforms using larger width to smooth the waveforms.
r4<-decom(xx,smooth=TRUE,width=5) ##when you change to a larger width, it can work,
```

```
                                  #but give you some unreasonable estimates, return NA

###original result from this decom is (you will not see it, the function filter this result
###and put NA for the estimation since they maybe not right results)
#Nonlinear regression model
#model:y~A1*exp(-(x-u1)^2/(2*sigma1^2))+A2*exp(-(x-u2)^2/(2*sigma2^2)) n\
#+A3*exp(-(x-u3)^2/(2*sigma3^2))
#data: df
#A1      A2      A3      u1      u2      u3  sigma1  sigma2  sigma3
#228.709 -30.883  81.869  41.640  42.131  71.680  14.613   3.522   8.073
##A (ampilitude should not be negative)

r5<-decom(xx,width=10) ##this will work by smoothing the waveform
r6<-decom(xx,thres=0.1,width=5)  ##by adjusting width and thres of real peak, you may
                                 ##get a reasonable results

# for the whole dataset
dr<-apply(return,1,decom)
```

`decom.adaptive` *decom.adaptive*

Description

The function allows you to eatimate parameters charcterizing waveforms and to pave the way for generating waveform-based point cloud.

Usage

```
decom.adaptive(x, smooth = TRUE, thres = 0.22, width = 3)
```

Arguments

`x`	is a waveform with a index at the begining and followed with intensities.
`smooth`	is tell whether you want to smooth the waveform to remove some obvious outliers. Default is TRUE.
`thres`	is to determine if the detected peak is the real peak whose intensity should be higher than threshold*maximum intensity. Default is 0.22.
`width`	width of moving window.Default is 3, must be odd integer between 1 and n.This parameter ONLY work when the smooth is TRUE.

Value

A list contains estimates of A, u, sig and ri (rate parameter in adaotive Gaussian function) after decomposition.

References

Tan Zhou, and Sorin C. Popescu, 2017. Bayesian decomposition of full waveform LiDAR data with uncertainty analysis. Remote Sensing of Environment 200 (2017): 43-62.

Examples

```
##import return waveform data
data(return)
lr<-nrow(return)
ind<-c(1:lr)
return<-data.frame(ind,return)
x<-return[1,] ###must be a dataset including intensity with index at the beginning.
r1<-decom(x)

r2<-decom.adaptive(x)


# for the whole dataset
dr3<-apply(return[50:200, ],1,decom.adaptive)
```

`decom.weibull` *decom.weibull*

Description

The function allows you to fit waveforms with the Weibull function and get parameters estimated.

Usage

```
decom.weibull(x, smooth = TRUE, thres = 0.22, width = 3)
```

Arguments

`x`	is a waveform with a index at the begining and followed with intensities.
`smooth`	is tell whether you want to smooth the waveform to remove some obvious outliers. Default is TRUE.
`thres`	is to determine if the detected peak is the real peak whose intensity should be higher than threshold*maximum intensity. Default is 0.22.
`width`	width of moving window.Default is 3, must be odd integer between 1 and n.This parameter ONLY work when the smooth is TRUE.

Value

A list contains estimates of A (amplitude parameter), u (location parameter), sigma (scale parameter) and k (shape parameter) after decomposition. these parameters are different from the adaptive Gaussian and Gaussian decomposition.

References

Tan Zhou, and Sorin C. Popescu, 2017. Bayesian decomposition of full waveform LiDAR data with uncertainty analysis. Remote Sensing of Environment 200 (2017): 43-62.

Examples

```
##import return waveform data
data(return)
lr<-nrow(return)
ind<-c(1:lr)
return<-data.frame(ind,return)
x<-return[1,] ###must be a dataset including intensity with index at the beginning.
r1<-decom.weibull(x)

r2<-decom.weibull(return[2,])


# for the whole dataset
dr3<-apply(return,1,decom.weibull)
dd<-return[10:20,]
dr4<-apply(dd,1,decom.weibull)
```

`decom_result` *Results of decomposition using Gaussian model*

Description

This dataset contained the result from the Gaussian decompositon.

Usage

```
decom_result
```

Format

A data frame with 1000 rows and 7 variables

index index of waveform, which can tell which rows or results belongs to specific waveform

A the amplitude of one waveform componment (A)

u the time location corresponds to the amplitude for one waveform componment

sigma the echo width of one waveform componment

A_std the standard error of A, which can be used for uncertainty analysis

u_std the standard error of u, which can be used for uncertainty analysis

sig_std the standard error of sigma, which can be used for uncertainty analysis

deconvolution *deconvolution*

Description

The function allows you to deconvolve retrun waveform lidar data by using the system impulse response (it can be given by the data provider or measured with the return impulse response and its corresponding outgoing pluse) and corresponding outgoing waveform to obtain effective target positions. Two algorithms including Gold or Richardson-Lucy algorithms were available.

Usage

```
deconvolution(
  re,
  out,
  imp,
  imp_out = NULL,
  method = c("Gold"),
  np = 2,
  rescale = TRUE,
  small_paras = c(30, 2, 1.8, 30, 2, 1.8),
  large_paras = c(30, 3, 1.8, 40, 3, 1.8),
  imp_out_pars = c(20, 5, 1.8)
)
```

Arguments

re	is the return waveform.
out	is the outgoing waveform, which should have the same number of rows.
imp	is the return impulse reponse (this should come with the corresponding outgoing pluse) or the sytem impulse reponse (when the corresponding outgoing pulse for the impluse response is not available or the data vendors directly provide system impluse response).
imp_out	is the corresponding outgoing pulse of the return impulse response. Through devonvolution of return impulse response, we can obtain true system impluse repsonse if it wasn't provided by the data vendor.
method	two methods including Gold or Richardson-Lucy (RL) algorithms. method=c("Gold","RL"). Default is method=c("Gold").
np	is a threshold value which determines use small_paras (smaller iterations and repetitions in the deconvolution) or or large_paras (larger iterations and repetitionsin the deconvolution). Default is 2.
rescale	is determine whether you want to rescale the waveform intensity. Generally, we used the minimum intensity to conduct rescaling. Default is to implement rescale.

small_paras is the deconvolution parameters including iterations, repetitions and boost when condcuting the system impulse and outgoing pulse deconvolution algorithms with waveform componments is less than np or the waveform is less complicated with few noise.

large_paras is the deconvolution parameters including iterations, repetitions and boost for the system impulse and outgoing pulse deconvolution when the number of waveform componments is more than np. Generally when the waveform become more complicated, we should use larger iteration, repetitions. iterations: number of iterations (parameter in the Gold deconvolution algorithm) between boosting operations; repetitions number of repetitions of boosting operations. It must be greater or equal to one.So the total number of iterations is repetitions*iterations boosting coefficient/exponent. Applies only if repetitions is greater than one. Recommended range [1,2].

imp_out_pars is the deconvolution parameters for obtaining system impluse response using the impulse response and corresponding outgoing pulse. As same as the small_paras and large_parameters. This parameter is effective only when the imp_out is not NULL.

Value

The deconvovled waveform.

References

Zhou, Tan*, Sorin C. Popescu, Keith Krause, Ryan D. Sheridan, and Eric Putman, 2017. Gold-A novel deconvolution algorithm with optimization for waveform LiDAR processing. ISPRS Journal of Photogrammetry and Remote Sensing 129 (2017): 131-150. https://doi.org/10.1016/j.isprsjprs.2017.04.021

Examples

```
library(waveformlidar)
if (!require("rPeaks")) {
    install_github("tankwin08/rPeaks")
  }
data(return)
data(outg)
data(imp)  ##The impulse function is generally one for the whole study area or
data(imp_out)

re<-return[1,]
out<-outg[1,]
imp<-imp

dr<-deconvolution(re,out,imp)
dr1<-deconvolution(re,out,imp,method="RL")
dr2<-deconvolution(re,out,imp,method="RL",small_paras=c(20,2,1.8,20,2,2))

plot(dr,type="l")
lines(dr1,col="red")
```

```
lines(dr2,col="blue")
###some differences can be observed when you used different parameters.
##In the real application, you need to find optimized parameters suitable for your case.
## In addition, the accuracy of impulse function significantly affects results based on experiments.
```

`decon_result` *Results of using the deconvolution and decomposition method.*

Description

This dataset contained the results using deconvolution and then decompostion method as described in the Tan Zhou*, Sorin C. Popescu, Keith Krause, Ryan D. Sheridan, and Eric Putman, 2017. Gold-A novel deconvolution algorithm with optimization for waveform LiDAR processing. ISPRS Journal of Photogrammetry and Remote Sensing 129 (2017): 131-150. https://doi.org/10.1016/j.isprsjprs.2017.04.021

Usage

```
decon_result
```

Format

A data frame with 1000 rows and 7 variables

index index of waveform, which can tell which rows or results belongs to specific waveform

A the amplitude of one waveform componment (A)

u the time location corresponds to the amplitude for one waveform componment

sigma the echo width of one waveform componment

A_std the standard error of A, which can be used for uncertainty analysis

u_std the standard error of u, which can be used for uncertainty analysis

sig_std the standard error of sigma, which can be used for uncertainty analysis

---

`fslope` *fslope*

Description

The function allows you calculate the (1) front slope angle (FS, the angle from waveform beginning to the first peak which is assumed to be canopy returns) and (2)roughness of outermost canopy (ROUGH, the distnace from the waveform beginning to the first peak).

Usage

```
fslope(y, smooth = TRUE, thres = 0.22, width = 5, tr = 1)
```

Arguments

y	is a waveform with only intensities.
smooth	is tell whether you want to smooth the waveform to reduce the effect of some obvious noise. Default is TRUE.
thres	is to determine if the detected peak is the real peak whose intensity should be higher than threshold*maximum intensity. Default is 0.22.
width	width of moving window.Default is 3, must be integer between 1 and n.This parameter ONLY works when the smooth is TRUE.
tr	the temporal resolution of waveform.Default is 1 ns. Must be integer from 1 to n.

Value

return the front slope angle (FS, The angle from waveform beginning to the first peak which is assumed to be canopy returns) and roughness of outermost canopy (ROUGH, the distnace from the waveform beginning to the first peak).

Examples

```
data(return)
###for the one peak, this function returns not useful results.
#This function more related to vegetation and canopy.
y<-return[1,]
#default
yr<-fslope(y)

yy<-return[182,]
yyr<-fslope(yy)
```

---

gennls *gennls*

Description

The function allows you to prepare the formula and start values for Gaussian decomposition

Usage

```
gennls(A, u, sig)
```

Arguments

A	is the amplitude of waveform.
u	is the peak location of waveform.
sig	is the echo width of waveform.

Value

return a formula suitable for different number of waveform componments with Gaussian distribution.The number of componments needs to be determined by the users. Generally you need to use peakfind function to roughly estimate the number of waveform componments

Examples

```
A<-c(76,56,80);u<-c(29,40,67);sig<-c(4,3,2.5) ##these three should have the same length
fg<-gennls(A,u,sig)
##input formula for Gaussian decomposition
fgf<-fg$formula
###start values
fgs<-fg$start
```

---

geo *The reference geolocation of the return waveforms.*

Description

This dataset should have the same number of rows of the return waveforms (return). It is generally coming with waveform data and provided by the data provider. The number of rows should be the same as the return waveform or pulse (return).

Usage

```
geo
```

Format

A data frame with 500 rows and 17 variables.

index index to connect return waveform and geolocation

`x` Easting of first return

`y` Northing of first return

`z` Height of first return

`dx` the position change per nanasecond at x direction

`dy` the position change per nanasecond at y direction

`dz` the position change per nanasecond at z direction

`or` outgoing pulse reference bin location (leading edge 50th point of the outgoing pulse)

`fr` first return reference bin location (leading edge 50th point of the first return)

`V9` Easting of Return Bin0

`V10` Northing of Return Bin0

`V11` Height of Return Bin0

V12 the position change per nanasecond at x direction

V13 the position change per nanasecond at y direction

V14 the position change per nanasecond at z direction

V15 outgoing pulse peak bin location

V16 return bin0 location

`geotransform` *geotransform*

Description

The function allows you to convert parameters from decomposition to point cloud by using georeference data (generally it should come with waveform data). Detailed description of how to calculate can refer to Zhou, T., Popescu, S.C., Krause, K., Sheridan, R.D., Putman, E., 2017. Gold - a noveldeconvolution algorithm with optimization for waveform LiDAR processing.ISPRS Journal of Photogrammetry and Remote Sensing 129 (2017): 131-150. For the direct decomposition method, three kinds of position are calculated: leading edge, peak and trail edge. For the deconvolution and decomposition method, only the peak position is used to calculate the position. In additional, the uncertianty of the point cloud was provided based on detected peaks' 95 you need to preprocess data to the same format or have the same required information or datsets.

Usage

```
geotransform(decomp, geo, source = "decomposition")
```

Arguments

`decomp`	the object from the decomposition or after deconvolution and decomposition.
`geo`	the reference geolocation that is generally coming with waveform data and provided by the data provider.
`source`	is determined by input data. If estimated parameters are from dcomposition, source = "decomposition". Otherwise will be deconvolution and decomposition. Default is decomposition.

Value

A dataframe with columns. For the direct decompostion method, we will have 29 columns and for the deconvolution and decomposition method. 17 columns were generated: index,pi,t,sd,pise,tse,sdse,px,py,pz,uncerUXpeak,

`index`	The index of waveform.
`pi`	The estimated amplitude of an waveform componment.
`t`	The estimated peak location of an waveform componment.
`sd`	The estimated echo width of an waveform componment.
`pise`	The standard error of the estimated amplitude.

tse	The standard error of the estimated peak location.
sdse	The standard error of the estimated echo width.
px	Desired x position using peak locations.
py	Desired y position using peak locations.
pz	Desired y position using peak locations.
lowx	Desired x position using leading edge locations.
lowy	Desired y position using leading edge locations.
lowz	Desired z position using leading edge locations.
upx	Desired x position using trailing edge locations.
upy	Desired y position using trailing edge locations.
upz	Desired z position using trailing edge locations.
uncerUXpeak	Upper bound of 95th confidence interval of px.
uncerUYpeak	Upper bound of 95th confidence interval of py.
uncerUZpeak	Upper bound of 95th confidence interval of pz.
uncerLXpeak	Lower bound of 95th confidence interval of px.
uncerLYpeak	Lower bound of 95th confidence interval of py.
uncerLZpeak	Lower bound of 95thconfidence interval of pz.
uncerUXleading	Upper bound of 95th confidence interval of lowx.
uncerUYleading	Upper bound of 95th confidence interval of lowy.
uncerUZleading	Upper bound of 95th confidence interval of lowz.
uncerLXleading	Lower bound of 95th confidence interval of lowx.
uncerLYleading	Lower bound of 95th confidence interval of lowy.
uncerLZleading	Lower bound of 95th confidence interval of lowz.
rn	The number of return for each waveform.

By combining xyz, the users can get waveform-based point cloud using leading edge, peak and trail edge positions, and parameter uncertainty of the point cloud.

References

Zhou, Tan*, Sorin C. Popescu, Keith Krause, Ryan D. Sheridan, and Eric Putman, 2017. Gold-A novel deconvolution algorithm with optimization for waveform LiDAR processing. ISPRS Journal of Photogrammetry and Remote Sensing 129 (2017): 131-150. https://doi.org/10.1016/j.isprsjprs.2017.04.021

Examples

```
data(geo)
data(decom_result)
data(decon_result)
##used part of data to show the results
decomp<-decom_result[1:80,]
geo<-geo[1:80]
```

```
deconp<-decon_result[1:80]

##the follwoing steps are reuired to conduct the geotransformation,
##we need assign exactly same column names
geoindex=c(1:9,16)
colnames(geo)[geoindex]<-c("index","orix","oriy","oriz","dx","dy","dz","outref","refbin","outpeak")
##for decomposition results

geor<-geotransform(decomp,geo)

########for deconvolution and decomposition
decongeo<-geotransform(decomp=deconp,geo,source="deconvolution")
```

`ground.location` *ground.location*

Description

The function allows you to identify the possible ground location (time index) in the waveform. Generally, we assume the last echo or peak corresponding to the ground.

Usage

```
ground.location(
  y,
  smooth = TRUE,
  rescale = TRUE,
  thres = 0.2,
  width = 3,
  top = TRUE
)
```

Arguments

`y`	is the waveform intensities. If you have other information, you should delete these intensites before you run this function .
`smooth`	is tell whether you want to smooth the waveform to reduce the effect of some obvious noise. Default is TRUE.
`rescale`	is to determine whether you want to rescale the waveform intensity or not. Here we used the minimum intensity of each waveform to conduct rescaling. Default is using rescaling.
`thres`	is to determine if the detected peak is the real peak. The real peak's intensity should be higher than threshold*maximum intensity. Default is 0.22.
`width`	the width of moving window for smoothing.Default is 3, must be integer between 1 and n.This parameter ONLY work when the smooth is TRUE.
`top`	is to tell whether we calculate the ground time location from the top (where waveform starts or canopy) or from the bottom (where the waveform ends). Default is from the top.

Value

return the index of possible ground position of waveform.

Examples

```
data(return)
x<-return[125,]
#default
ground.location(x)
```

`hyperpointcloud` *hyperpointcloud*

Description

The function allows you to convert every waveform intensities into points which formed hyper point cloud. This can help you visualize the waveform data in an efficient way. It will generate a big dataset which may reach the memory of computer's RAM. Thus, it requires to split the study regions into smaller tiles when you have large study sites.

Usage

```
hyperpointcloud(waveform, geo)
```

Arguments

`waveform`	the raw waveform data.
`geo`	the reference geolocation that corresponds to the raw waveform data which requres to have the same number of rows as the waveform data.

Value

A dataframe with 4 columns including xyz geolocation and intensity.

`x`	The x position of one waveform intensity
`y`	The y position of one waveform intensity
`z`	The z position of one waveform intensity
`intensity`	The position's intensity

References

Zhou, Tan, Sorin Popescu, Lonesome Malambo, Kaiguang Zhao, and Keith Krause. From LiDAR waveforms to Hyper Point Clouds: a novel data product to characterize vegetation structure. Remote Sensing 10, no. 12 (2018): 1949.

Examples

```
data(return)  ###import raw return waveforms
data(geo)  ###import corresponding reference geolocation
geo$index<- NULL
colnames(geo)[1:8]<- c("x","y","z","dx","dy","dz","or","fr")
### you should know which columns corresponding to above column names
### before run the hyperpointcloud when you used your own new datasets
hpr<- hyperpointcloud (waveform = return,geo = geo)
```

`imp` *prototype system impulse*

Description

This data was generated from a hard ground target with a mirror angle close to nadir and its corresponding outgoing waveform. Generally one study site only need one system impulse which is mainly for calibration purpose.

Usage

```
imp
```

Format

A data frame with 100 rows and 1 column.

V1 The intensity of the system impulse

`imp_out` *Outgoing pulse corresponding to the system impulse*

Description

This dataset is the corresponding outgoing pulse of the system impulse

Usage

```
imp_out
```

Format

A vector with the length of 100. The zero padding was used to make sure every waveforms has the same length. In this case, 0 means no recorded intensity.

outgoing pulse

integral *integral*

Description

The function allows you to calculate the integral of intensity from ground part, vegetation part, sum of them and ratio between vegation integral and total integral with user-defined vegetation and ground boundary.

Usage

```
integral(
  y,
  smooth = TRUE,
  rescale = TRUE,
  thres = 0.2,
  width = 3,
  tr = 1,
  dis = 20
)
```

Arguments

y	is the waveform intensities. If you have other information, you should delete these intensites before you run this function .
smooth	is tell whether you want to smooth the waveform to reduce the effect of some obvious noise. Default is TRUE.
rescale	is to determine whether you want to rescale the waveform intensity or not. Here we used the minimum intensity of each waveform to conduct rescaling. Default is using rescaling.
thres	is to determine if the detected peak is the real peak whose intensity should be higher than threshold*maximum intensity. Default is 0.2.
width	width of moving window.Default is 3, must be integer between 1 and n.This parameter ONLY work when the smooth is TRUE.
tr	the temporal resolution of waveform.Default is 1 ns. Must be integer from 1 to n.
dis	the distance from last echo (assumed ground) which determine the boundary between ground and vegetation. Default is 20 ns which equals to 3 m (20*0.15*1).This means the ground part signals are from the assumed ground location to 20 ns above or the signals of vegetation are from waveform begining to 3 m above ground.

Value

return the integral of waveform intensity for the ground, vegetation parts, the whole waveform above ground and the integral ration betwwen vegetation and the whole waveform .

Examples

```
data(return)
x<-return[1,]
##if we kept everything as default, before you use this function, you need to know the
## temporal resolution of waveform, default is 1 ns.

##for one peak, it generaly not make too much sense since generally only ground was
##present in this case.

r1<-integral(x)

#if you didn't want to smooth,
r2<-integral(x,smooth=FALSE)

##you also can define the boundary between vegetation and ground by assign adjusting dis
#if we assign 15, it means vegetation starts at 2.25 (15*1*0.15) m above the last echo.

r3<-integral(x,dis=15)
# when it comes to the waveform with several peaks
xx<-return[182,]

rr1<-integral(xx)
```

---

lpeak *lpeak*

Description

This function allows you to locate where are peaks of a waveform and return a list of TRUE and FALSE TRUE represents this location is the peak.

Usage

```
lpeak(series, span = 3)
```

Arguments

series	is the input a numeric vector.
span	is the length or interval of peak finding cell, default is 3.

Value

return a boolean type data corresponding to the numeric vector.

Examples

```
data(return)
w1<-return[1,]
#w1 is numeric values represnt an waveform.
lpeak(w1,3)
w2<-return[3,]
lpeak(w2,5)
```

---

maxamp *maxamp*

Description

The function allows you to find the maximum amplitude of waveform above the ground (not the whole waveform) if this waveform come from vegetation with more than 1 peaks. Otherwise it will give the last peak's corresponding intensity. The identified maximum intensity can be used to calculate the height from waveform ending or waveform begining.

Usage

```
maxamp(y, smooth = TRUE, thres = 0.2, width = 3)
```

Arguments

y	is the waveform intensities. If you have other information, you should delete these intensites before you run this function .
smooth	is tell whether you want to smooth the waveform to reduce the effect of some obvious noise. Default is TRUE.
thres	is to determine if the detected peak is the real peak whose intensity should be higher than threshold*maximum intensity. Default is 0.2.
width	width of moving window.Default is 3, must be integer between 1 and n.This parameter ONLY work when the smooth is TRUE.

Value

return the largest amplitude above the ground for the waveform with more than two peaks and ground amplitude for the waveform with one peak.

Examples

```
data(return)
x<- return[1,]

rx<- maxamp(x)
###for more complicated waveforms
x1<- return[182,]
rx1<- maxamp(x1)
```

med.height *med.height*

Description

The function allows you to calculate the height of half total energy above ground (not total waveform) and height proportion at the half energy position. If you are interested in variables from the total waveform, you can use which.half or percentile.location functions.

Usage

```
med.height(x, smooth = TRUE, thres = 0.2, width = 3, tr = 1)
```

Arguments

x	is the waveform intensities. If you have other information, you should delete these intensites before you run this function .
smooth	is tell whether you want to smooth the waveform to reduce the effect of some obvious noise. Default is TRUE.
thres	is to determine if the detected peak is the real peak whose intensity should be higher than threshold*maximum intensity. Default is 0.22.
width	width of moving window.Default is 3, must be integer between 1 and n.This parameter ONLY work when the smooth is TRUE.
tr	the temporal resolution of waveform.Default is 1 ns. Must be integer from 1 to n.

Value

return the height of half total energy above the ground and the height proportion of the first half (from waveform begining to the half total energy position).

Examples

```
data(return)
x<-return[1,]
#default
med.height(x)  ###for one peak waveform, it generally not make too much sense
xx<-return[182,]
med.height(xx)  ##this can help to charcterize the waveform
```

npeaks *npeaks*

Description

The function allows you to briefly know how many peaks you have in a waveform

Usage

```
npeaks(y, drop = c(0, 0), smooth = TRUE, threshold = 0.2)
```

Arguments

y	is the waveform intensities.
drop	is the index of waveform index we should ingore or non-intensity of waveform information.Default is c(0,0) that means use the all input data.
smooth	is tell whether you want to smooth the waveform to reduce the effect of some obvious noise. Default is TRUE.
threshold	is to determine if the detected peak is the real peak whose intensity should be higher than threshold*maximum intensity. Default is 0.2.

Value

return the number of waveform componments.

Examples

```
##import return waveform data
data(return)
###individual waveform
  x<-return[1,]
  npeaks(x)
  npeaks(x,smooth=FALSE) ##it will use the raw data to detect peaks
  #you can set up threshold to determine if peaks are correctly identified.
  npeaks(x,smooth=FALSE,threshold=0.25)
###if there are some columns are not intensity, you can delete before you process
   y<-c(c(1,2,3),as.numeric(x))
   npeaks(y,drop=c(1,3))
```

outg *500 corresponding outgoing pulses.*

Description

This dataset should have the same number of rows as the return pulses, which is the outgoing pulse for each return waveform pulse.The zero padding was used to make sure every waveforms has the same length. In this case, 0 means no recorded intensity.

Usage

```
outg
```

Format

A data frame with 500 rows and 100 variables:

peakfind *peakfind*

Description

The function allows you to roughly estimate A,u,sig parameters will fit in the Gaussian decomposition.

Usage

```
peakfind(x, smooth = TRUE, thres = 0.2, width = 3)
```

Arguments

x	is a waveform with a index at the begining mainly to .
smooth	is tell whether you want to smooth the waveform to reduce the effect of some obvious noise. Default is TRUE.
thres	is to determine if the detected peak is the real peak whose intensity should be higher than threshold*maximum intensity. Default is 0.2.
width	width of moving window.Default is 3, must be integer between 1 and n. This parameter ONLY work when the smooth is TRUE.

Value

return a list contains waveform index, rough estimates of A, u, sig.

References

Tan Zhou, and Sorin C. Popescu, 2017. Bayesian decomposition of full waveform LiDAR data with uncertainty analysis. Remote Sensing of Environment 200 (2017): 43-62.

Examples

```
##import return waveform data
data(return)
ind<-c(1:nrow(return))
return<-data.frame(ind,return)
x<-return[1,] ###must be must be a dataset incluing intensity with index at the beginning.
peakfind(x) ## index, estimated A, u, and sig

##to get accurate estimates of A, u,g, you need to explore your dataset to optimized parameters.
##generally thres affects a lot, assigning smooth to TRUE is preferable in most of cases.
#for the whole dataset
dr<-apply(return,1,peakfind)
####to manage data and store in a data frame.
rpf<-do.call("rbind",lapply(dr,"[[",1))
```

`percentile.location` *percentile.location*

Description

The function allows you to calculate the time location position of energy percentile based on the given quantile within a waveform. With this information. In addition, you can calculate the energy percentile heights. Here, the percentile will be calculated from the top (more likely the canopy part) by defalut.

Usage

```
percentile.location(x, quan = seq(0.5, 1, 0.1), rescale = TRUE, top = TRUE)
```

Arguments

`x`	is the waveform intensities. If you have other information, you should delete these intensites before you run this function .
`quan`	is the quantile of energy you are interested in. Defalut is seq(0.5,1,0.1).
`rescale`	is to determine whether you want to rescale the waveform intensity or not. Here we used the minimum intensity of each waveform to conduct rescaling. Default is using rescaling.
`top`	is to tell whether we calculate the percentile from the top (where waveform starts or canopy) or from the bottom (where the waveform ends). Default is from the top.

Value

return the index or position of corressponding quantile energy of waveform starting from the top.

Examples

```
data(return)
x<-return[1,]
#default
percentile.location(x)

#change the quantile
qr<-seq(0.45,0.99,0.05)
re1<-percentile.location(x,quan=qr)
###rescale affects your results
re2<-percentile.location(x,quan=qr,rescale = FALSE)
##after you get the index, you can convert them to the height
#based on temporal resolution or georeference information of this waveform.
```

`rawtocomposite` *rawtocomposite*

Description

The function allows you to convert point cloud after waveformvoxel or raw waveforms into composite waveforms (with vertical distribution of intensity) by reducing the effect of off-naid angle of emitted laser. The conversion is based on the waveform voxellization product. Four kinds of values you can chose to represent the intensity of composite waveform: the number of intensity (generally is not useful), mean intensity, maximum intensity and total intensity (the last one is also not prefered in most of cases).

Usage

```
rawtocomposite(voxr, inten_index = 2)
```

Arguments

`voxr`	the object from the waveformvoxel.
`inten_index`	the value (1,2,3,4,...) to represnt the intensity of composite waveforms.It is a integer from 1 to 4 and default is 2. 1: the number of intensity of the voxel (generally is not useful); 2: the maximum intensity of the waveform voxel; 3: the mean intensity of the waveform voxel; 4: the total intensity of voxel(the last one is also not prefered in most cases)

Value

A dataframe with first three columns including geolocation xyz of the first Non-NA intensity (Highest position) and intensities along the height bins, other non-NA values are intensities for the rest columns.

x	The x position of the first Non-NA intensity or highest intensity position in one waveform
y	The y position of the first Non-NA intensity or highest intensity position in one waveform
z	The z position of the first Non-NA intensity or highest intensity position in one waveform
intensity 1	The intnesity of first height bin
intensity 2	The intensity of second height bin
...	Intensities along the height bin

Examples

```
data(return)  ###import raw return waveforms
data(geo)  ###import corresponding reference geolocation
colnames(geo)[2:9]<-c("x","y","z","dx","dy","dz","or","fr")
### you should know which columns corresponding to above column names before
### run the hyperpointcloud when you used your own new datasets.

hpr<-hyperpointcloud(waveform=return,geo=geo)

##beofre run waveformvoxel, we need to create hyperpointcloud first
##this exampel we just used 100000 points to reduce processing time

voxr<-waveformvoxel(hpc=hpr,res=c(1,1,0.3))
rtc<-rawtocomposite(voxr)
```

---

return

500 sample LiDAR waveforms from Harvard Forest provided by National Ecological Observatory Networks (NEON). More details can be found in Tan Zhou, Sorin C. Popescu, Keith Krause, Ryan D. Sheridan, and Eric Putman, 2017. Gold-A novel deconvolution algorithm with optimization for waveform LiDAR processing. ISPRS Journal of Photogrammetry and Remote Sensing 129 (2017): 131-150. https://doi.org/10.1016/j.isprsjprs.2017.04.021*

Description

A dataset containing the return waveforms with intensity along the path. The zero padding was used to make sure every waveforms has the same length. In this case, 0 means no recorded intensity.

Usage

```
return
```

Format

A data frame with 500 rows and 208 variables:

---

shp_hf *A boundary shapefile, polygon*

Description

This data is manily to test the fucntion waveformclip.

Usage

```
shp_hf
```

Format

A shapefile containing a small boudary of the Harvard Forest.

---

waveformclip *waveformclip*

Description

The function allows you to select waveforms of your interest from the whole waveform dataset based on the shpfile(s) or geoextent including xmin,xmax,ymin,ymax.

Usage

```
waveformclip(waveform, geo, shp, geoextent = c())
```

Arguments

waveform	The raw waveform data containing intensities and index of waveform.
geo	The reference geolocation that corresponds to the raw waveform data which has the same number of rows as the waveform data.
shp	The region of interst which tells the regions to clip, it require has the same projetct coordinate system (UTM is prefered) as the geo data.
geoextent	Another way to clip waveform using xmin, xmax, ymin, ymax. Please note you should specify the parameter in this order. Defalut is NULL.

Value

For using shapefile, it will return a dataframe with shapefile index (this will be useful for multipolygons) and corresponding select waveforms in each sub shapefiles. For the geoextent, it will return the selected waveforms in the extent and corresponind selected index of geo data.

Examples

```
data(return)  ###import raw return waveforms
data(geo)  ###import corresponding reference geolocation
data(shp_hf)  ###import shpefile
##the next step is required, since everybody's georeference data maybe
##a bit difference, you need to adjust by yourself when you implement the function.

### you need to assign x and y columns at least to make the function run properly
colnames(geo)[2:9]<-c("x","y","z","dx","dy","dz","or","fr")

##use shp file
##this step is required. Mainly to identify index of selected waveforms from original datasets
waveform<-cbind(waveformindex=1:nrow(return),return)
geo<-geo
shp<-shp_hf
swre<-waveformclip(waveform,geo,shp)

###use geoextent
swre1<-waveformclip(waveform,geo,geoextent=c(731126,731128,4712678,4712698))
```

`waveformgrid` *waveformgrid*

Description

The function allows you to project raw waveforms into 2d grids (can be xy, yz, and xz) with self defined resolution. For the values of each grid, four kinds of values were available to be used: the total number of intensity in each grid, maximum intensity of the grid, mean intensity of the grid, and total intensity of the grid.

Usage

```
waveformgrid(
  hpc,
  waveform = NULL,
  geo = NULL,
  quan = NULL,
  res = c(0.8, 0.8),
  method = "HPC"
)
```

Arguments

hpc	The hyper point cloud generated from the raw waveform data with hyperpointcloud function or the data with the same structure as the hyperpointcloud object.
waveform	The raw waveform data with only intensities.
geo	The reference geolocation that corresponds to the raw waveform data which requres has the same row as the waveform data. At least three colomns should be assigned: original x, y and first return reference bin location (fr,leading edge 50 You can see detailed description of the columns in geotransform.
quan	The quantile of intenisty in the given grid or voxel. Defalut is NULL.
res	The grid size with x and y spatial resolution. Default is 0.8*0.8, res=c(0.8,0.8).
method	There are two methods c("HPC","Other") to project the waveforms into the 2d surface. Default ("HPC") is to use the HPC as the input data. The waveform and geo will be NULL. "Other" is to use the raw waveform and corresponding reference geolcation data to generate the intensities in each grid, and hpc should be NULL.

Value

A dataframe with 7 columns including geolocation (center of the grid) and intensities. Specifically, these 7 columns are "index","cx","cy","length_maxi_meani_totali.1","length_maxi_meani_totali.2", "length_maxi_meani_totali.3","length_maxi_meani_totali.4").

index	The index of the grid
cx	The mean x of the grid center
cy	The mean y of the grid center
intensity.length or value.length	The number of intensity in the grid. intensity.length is for the HPC method and value.intensity for the others
intensity.maxi or value.maxi	The maximum intensity of waveforms in the grid
intensity.meani or value.meani	The mean intensity of waveforms in the grid
intensity.totali or value.totali	The total intensity of waveforms in the grid
...	Percentile intensity based on the quan

By assigning differnt columns as x and y, the users can project waveforms into the xy, xz, yz surfaces.

References

Tan Zhou, Sorin Popescu, Lonesome Malambo, Kaiguang Zhao, Keith Krause. From LiDAR waveforms to Hyper Point Clouds: a novel data product to characterize vegetation structure. Remote Sensing 2018, 10(12), 1949; https://doi.org/10.3390/rs10121949

Examples

```
data(return)  ###import raw return waveforms
data(geo)  ###import corresponding reference geolocation
##if your geo data didn't the same column names as the follwoing one, you need to change it to
#it to the same column names. One thing you need to figure out is the corresponding column numbers
colnames(geo)[1:8]<-c("x","y","z","dx","dy","dz","or","fr")
###at least you should know which columns corresponding to x,y and fr before run the waveformgrid
##using the raw data
grid_re<-waveformgrid(waveform=return,geo=geo,res=c(0.8,0.8),method = "Other")

##using the hpc object
hpc<-hyperpointcloud(waveform=return,geo=geo)

hpcgrid<-waveformgrid(hpc=hpc,res=c(1,1))
```

`waveformvoxel` *waveformvoxel*

Description

The function allows you to project raw waveforms into 3d voxels from hyperpointcloud with self defined resolution. For the intensity of each grid, four kinds of values were available to be used: the total number of intensity in each voxel, maximum intensity of the voxel, mean intensity of the voxel, and total intensity of the voxel.

Usage

```
waveformvoxel(hpc, res = c(0.8, 0.8, 0.15), quan = NULL)
```

Arguments

`hpc`	the objetcs from hyperpointcloud function.
`res`	the voxel size with x, y, and z spatial resolution. Default is 0.8*0.8*0.15.
`quan`	The quantile of intenisty in the given grid or voxel. Defalut is NULL.

Value

A dataframe with 7 columns including geolocation and intensities. Specifically,"index","cx","cy","length_maxi_meani_totali "length_maxi_meani_totali.3","length_maxi_meani_totali.4").

`index`	The index of the voxel
`cx`	The average x of the voxel center
`cy`	The average y of the voxel center
`cz`	The average z of the voxel center
`intensity.length`	The number of intensity in the voxel

`intensity.maxi` The maximum intensity of waveforms in the voxel

`intensity.meani`
The mean intensity of waveforms in the voxel

`intensity.totali`
The total intensity of waveforms in the voxel

`...` Percentile intensity based on the quan

References

Tan Zhou, Sorin Popescu, Lonesome Malambo, Kaiguang Zhao, Keith Krause. From LiDAR waveforms to Hyper Point Clouds: a novel data product to characterize vegetation structure. Remote Sensing 2018, 10(12), 1949; https://doi.org/10.3390/rs10121949

Examples

```
data(return)  ###import raw return waveforms
data(geo)  ###import corresponding reference geolocation

#' ### you should know which columns corresponding to above column names
## before run the hyperpointcloud when you used your own new datasets, this is very important step
colnames(geo)[2:9]<-c("x","y","z","dx","dy","dz","or","fr")

hpr<- hyperpointcloud(waveform = return,geo = geo)

##beofre run waveformvoxel, we need to create hyperpointcloud first
##this exampel we just used 100000 points to reduce processing time

voxr<-waveformvoxel(hpc = hpr,res=c(1,1,0.3))
```

`wavelen` *wavelen*

Description

The function allows you to measure the length of the waveform with intensity after excluding non-intensity parts. Generally, we assumed the intensity of a waveform should be larger than 0.

Usage

```
wavelen(x)
```

Arguments

`x` is the waveform intensities. If you have other information, you should delete these intensites before you run this function .

Value

return the the number of waveform intensities or waveform length.

Examples

```
data(return)
y<-cbind(1:nrow(return),return)
x<-return[1,]
wavelen(x)
#for x has index and intensities, we should delete non-intensities first
yy<-y[,-1]
xx<-yy[1,]
wavelen(xx)
```

---

wgennls *wgennls*

Description

The function allows you to prepare the formula and start values for the decomposition using Weibull functions.

Usage

```
wgennls(A, u, sig, k)
```

Arguments

A	is the amplitude of waveform.
u	is a location parameter in the Weibull model, but this one is different from Gaussian or adaptive Gaussian distribution.
sig	the scale parameter that controls the spread of the distribution,sig>0.
k	is the shape parameter that controls the behaviour or the shape of distribution, k>0.

Value

A formula suitable for different number of waveform componments with Weibull distribution.

Examples

```
A<-c(1000,900,1500);u<-c(-3,-5,0);sig<-c(30,40,75); k<-c(3,3,3)
##these four should have the same length
fg<-wgennls(A,u,sig,k)
##input formula for Gaussian decomposition
fgf<-fg$formula
```

```
###start values
fgs<-fg$start
```

---

which.half *which.half*

Description

The function allows you to identify the half total energy position (time location) which can be used to calculate the height from waveform ending or waveform begining.

Usage

```
which.half(x, rescale = TRUE)
```

Arguments

x	is the waveform intensities. If you have other information, you should delete these intensites before you run this function.
rescale	is to determine whether you want to rescale the waveform intensity or not. Here we used the minimum intensity of each waveform to conduct rescaling. Default is using rescaling.

Value

return the index of half total energy position of waveform.

Examples

```
data(return)
x<-return[1,]
#default
which.half(x)
#NOT USE rescale, the result will be a bit different
half_pos <- which.half(x,rescale=FALSE)
##the distnace from half position to the waveform ending is
dis_end = (wavelen(x)-half_pos)*0.15
dis_begin = half_pos*0.15 ##here 0.15 is the 1ns distance.
```

Index