# Package ‘rGEDI’

April 6, 2020

**Type** Package

**Title** NASA’s Global Ecosystem Dynamics Investigation (GEDI) Data Visualization and Processing

**Version** 0.1.6

**Maintainer** Carlos Alberto Silva <carlos_engflorestal@outlook.com>

**Description** Set of tools for downloading, reading, visualizing and processing GEDI Level1B, Level2A and Level2B data.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** methods

**Imports** bit64, curl, data.table, fs, getPass, ggplot2, hdf5r, jsonlite, lazyeval, raster, RColorBrewer, sp, stats

**Suggests** devtools, knitr, leaflet, leafsync, lidR, plot3D, rasterVis, rmarkdown, viridis, xtable

**SystemRequirements** GNU Scientific Library (>= 2.1), HDF5 (>= 1.8.13), libgeotiff (>= 1.4.0), szip (>= 2.1) zlib (>= 1.2)

**NeedsCompilation** yes

**RoxygenNote** 7.1.0

**VignetteBuilder** knitr

**BugReports** https://groups.yahoo.com/neo/groups/rGEDI

**URL** https://github.com/carlos-alberto-silva/rGEDI

**Copyright** inst/COPYRIGHTS

**Author** Carlos Alberto Silva [aut, cre, cph], Caio Hamamura [aut, cph], Ruben Valbuena [aut, ctb], Steven Hancock [aut, ctb], Adrian Cardil [aut, ctb], Eben North Broadbent [aut, ctb], Danilo Roberti Alves de Almeida [aut, ctb],
Celso H. L. Silva Junior [aut, ctb],
Carine Klauberg [aut, ctb],
Burton Garbow [cph] (Is the author of the MINPACK-1 Least Squares Fitting Library),
Kenneth Hillstrom [cph] (Is the author of the MINPACK-1 Least Squares Fitting Library),
Jorge More [cph] (Is the author of the MINPACK-1 Least Squares Fitting Library),
Craig Markwardt [cph] (Is the author of the enhanced MINPACK-1 Least Squares Fitting Library)

Repository  CRAN
Repository/R-Forge/Project  rgedi
Repository/R-Forge/Revision  25
Repository/R-Forge/DateTimeStamp  2020-04-04 19:49:33
Date/Publication  2020-04-06 09:10:02 UTC

\textbf{R topics documented:}

\begin{itemize}
  \item rGEDI-package  \hspace{1cm} 3
  \item clipLevel1B  \hspace{1cm} 4
  \item clipLevel1BGeo  \hspace{1cm} 5
  \item clipLevel1BGeoGeometry  \hspace{1cm} 6
  \item clipLevel1BGeometry  \hspace{1cm} 8
  \item clipLevel2A  \hspace{1cm} 9
  \item clipLevel2AGeometry  \hspace{1cm} 10
  \item clipLevel2AM  \hspace{1cm} 12
  \item clipLevel2AMGeometry  \hspace{1cm} 13
  \item clipLevel2B  \hspace{1cm} 14
  \item clipLevel2BGeometry  \hspace{1cm} 16
  \item clipLevel2BPAIProfile  \hspace{1cm} 17
  \item clipLevel2BPAIProfileGeometry  \hspace{1cm} 18
  \item clipLevel2BPAVDProfile  \hspace{1cm} 20
  \item clipLevel2BPAVDProfileGeometry  \hspace{1cm} 21
  \item clipLevel2BVPM  \hspace{1cm} 22
  \item clipLevel2BVPMGeometry  \hspace{1cm} 24
  \item close  \hspace{1cm} 25
  \item gedi.fullwaveform-class  \hspace{1cm} 26
  \item gedi.level1b-class  \hspace{1cm} 26
  \item gedi.level1bSim-class  \hspace{1cm} 27
  \item gedi.level2a-class  \hspace{1cm} 27
  \item gedi.level2b-class  \hspace{1cm} 28
  \item gediDownload  \hspace{1cm} 28
  \item gedifinder  \hspace{1cm} 29
  \item gediWFMetrics  \hspace{1cm} 31
  \item gediWFSimulator  \hspace{1cm} 38
  \item getLevel1BGeo  \hspace{1cm} 41
\end{itemize}
The rGEDI package provides functions for i) downloading, ii) visualizing, iii) clipping, iv) gridding, iv) simulating and v) exporting GEDI data.

Note

See more details about GEDI data in https://gedi.umd.edu/data/products/.

Author(s)

Carlos A. Silva, Caio Hamamura, Ruben Valbuena, Steve Hancock, Adrian Cardil, Eben N. Broadbent, Danilo R. A. de Almeida, Celso H. L. Silva Junior and Carine Klauberg

See Also

For comprehensive examples refer to vignette("Tutorial", package = "rGEDI")
Clip GEDI Level1B data by Coordinates

Description

This function clips GEDI Level1B data (geolocated waveforms) within a given bounding coordinates.

Usage

clipLevel1B(level1b, xmin, xmax, ymin, ymax, output)

Arguments

- level1b: A GEDI Level1B object (output of readLevel1B function). An S4 object of class "gedi.level1b".
- xmin: Numeric. West longitude (x) coordinate of the bounding rectangle, in decimal degrees.
- xmax: Numeric. East longitude (x) coordinate of the bounding rectangle, in decimal degrees.
- ymin: Numeric. South latitude (y) coordinate of the bounding rectangle, in decimal degrees.
- output: Optional character path where to save the new hdf5 file. The default stores a temporary file only.

Value

Returns a list of S4 objects of class "gedi.level1b" containing clipped GEDI Level1B data.

See Also

https://lpdaac.usgs.gov/products/gedi01_bv001/

Examples

# Specifying the path to GEDI level1B data (zip file)
outdir = tempdir()

level1B_fp_zip <- system.file("extdata",
"GEDI01_B_2019108080338_001964_T05337_02_003_01_sub.zip",
package="rGEDI")

# Unzipping GEDI level1B data
level1Bpath <- unzip(level1B_fp_zip,exdir = outdir)
# Reading GEDI level1B data (h5 file)
level1b<-readLevel1B(level1Bpath=level1Bpath)

# Bounding rectangle coordinates
xmin=-44.13
xmax=-44.12
ymin=-13.74
ymax=-13.73

# Specifying output file and path
output<-file.path(outdir,"GEDI01_B_2019100808338_001964_T05337_02_003_01_clip")

# Clipping GEDI Level1B data by extent boundary box
level1b_clip <- clipLevel1B(level1b,xmin, xmax, ymin, ymax, output)

close(level1b)
close(level1b_clip)

---

**clipLevel1BGeo**

*Clip GEDI Full Waveform Geolocations by Coordinates*

**Description**

This function clips GEDI level1B extracted geolocation (level1BGeo) data a within given bounding coordinates

**Usage**

clipLevel1BGeo(level1BGeo, xmin, xmax, ymin, ymax)

**Arguments**

- `level1BGeo` A GEDI Level1b object (output of `readLevel1B` function). An S4 object of class "gedi.level1b".
- `xmin` Numeric. West longitude (x) coordinate of the bounding rectangle, in decimal degrees.
- `xmax` Numeric. East longitude (x) coordinate of the bounding rectangle, in decimal degrees.
- `ymin` Numeric. South latitude (y) coordinate of the bounding rectangle, in decimal degrees.
- `ymax` Numeric. North latitude (y) coordinate of the bounding rectangle, in decimal degrees.

**Value**

Returns an S4 object of class `data.table-class`. 
See Also

https://lpdaac.usgs.gov/products/gedi01_bv001/

Examples

```r
# Specifying the path to GEDI level1B data (zip file)
outdir = tempdir()
level1B_fp_zip <- system.file("extdata",
  "GEDI01_B_2019100800338_001964_T05337_02_003_01_sub.zip",
  package="rGEDI")

# Unzipping GEDI level1B data
level1Bpath <- unzip(level1B_fp_zip, exdir = outdir)

# Reading GEDI level1B data (h5 file)
level1b <- readLevel1B(level1Bpath = level1Bpath)

# Extracting GEDI Full Waveform Geolocations
level1bGeo <- getLevel1BGeo(level1b)

# Bounding rectangle coordinates
xmin = -44.15036
xmax = -44.10066
ymin = -13.75831
ymax = -13.71244

# Clipping GEDI Full Waveform Geolocations by boundary box extent
level1bGeo_clip <- clipLevel1BGeo(level1bGeo, xmin, xmax, ymin, ymax)

library(leaflet)
leaflet() %>%
  addCircleMarkers(level1bGeo_clip$longitude_bin0,
  level1bGeo_clip$latitude_bin0,
  radius = 1,
  opacity = 1,
  color = "red") %>%
  addScaleBar(options = list(imperial = FALSE)) %>%
  addProviderTiles(providers$Esri.WorldImagery)

close(level1b)
```

---

**clipLevel1BGeoGeometry**

*Clip GEDI Full Waveform Geolocations by geometry*

**Description**

This function clips GEDI level1B extracted geolocation (level1BGeo) data within a given geometry.
clipLevel1BGeoGeometry

Usage

clipLevel1BGeoGeometry(level1BGeo, polygon_spdf, split_by)

Arguments

level1BGeo A GEDI Level1b object (output of readLevel1B function). An S4 object of class "data.table".

polygon_spdf Polygon. An object of class SpatialPolygonsDataFrame-class, which can be loaded as an ESRI shapefile using raster::shapefile() function in the raster package.

split_by Polygon id. If defined, GEDI data will be clipped by each polygon using the polygon id from table of attribute defined by the user.

Value

Returns an S4 object of class data.table-class containing the clipped GEDI level1B extracted geolocations.

See Also

https://lpdaac.usgs.gov/products/gedi01_bv001/

Examples

# Specifying the path to GEDI level1B data (zip file)
outdir = tempdir()
level1B_fp_zip <- system.file("extdata",
    "GEDI01_B_2019108080338_001964_T05337_02_003_01_sub.zip",
    package="rGEDI")

# Unzipping GEDI level1B data
level1Bpath <- unzip(level1B_fp_zip,exdir = outdir)

# Reading GEDI level1B data (h5 file)
level1b<-readLevel1B(level1Bpath=level1Bpath)

# Extracting GEDI Full Waveform Geolocations
level1bGeo<-getLevel1BGeo(level1b)

# Specifying the path to shapefile
polygon_filepath <- system.file("extdata", "stands_cerrado.shp", package="rGEDI")

# Reading shapefile as SpatialPolygonsDataFrame object
library(raster)
polygon_spdf<-shapefile(polygon_filepath)

# Clipping GEDI Full Waveform Geolocations by Geometry
level1bGeo_clip = clipLevel1BGeoGeometry(level1bGeo, polygon_spdf, split_by="id")

library(leaflet)
clipLevel1BGeometry

Clip GEDI Level1B data by geometry

Description

This function clips GEDI Level1B (geolocated waveforms) data within a given bounding geometry.

Usage

clipLevel1BGeometry(level1b, polygon_spdf, output = "", split_by = NULL)

Arguments

level1b A GEDI Level1B object (output of readLevel1B function). An S4 object of class "gedi.level1b".

polygon_spdf Polygon. An object of class SpatialPolygonsDataFrame-class, which can be loaded as an ESRI shapefile using raster::shapefile() function in the raster package.

output Optional character path where to save the new hdf5 file. The default stores a temporary file only.

split_by Polygon id. If defined, GEDI data will be clipped by each polygon using the attribute specified by split_by from the attribute table.

Value

Returns a list of S4 object of class "gedi.level1b" containing clipped GEDI Level1B data.

Examples

outdir = tempdir()

# Specifying the path to GEDI level1B data (zip file)
level1B_fp_zip <- system.file("extdata", "GEDI01_B_2019108080338_001964_T05337_02_003_01_sub.zip", 

```r
leaflet() %>%
  addCircleMarkers(level1bGeo_clip$longitude_bin0,
  level1bGeo_clip$latitude_bin0,
  radius = 1,
  opacity = 1,
  color = "red") %>%
  addScaleBar(options = list(imperial = FALSE)) %>%
  addPolygons(data=polygon_spdf,weight=1,col = 'white',
  opacity = 1, fillOpacity = 0) %>%
  addProviderTiles(providers$Esri.WorldImagery)

close(level1b)
```
# Unzipping GEDI level1B data
level1Bpath <- unzip(level1B_fp_zip, exdir = outdir)

# Reading GEDI level1B data (h5 file)
level1b <- readLevel1B(level1Bpath = level1Bpath)

# Specifying the path to shapefile
polygon_filepath <- system.file("extdata", "stands_cerrado.shp", package = "rGEDI")

# Reading shapefile as SpatialPolygonsDataFrame object
library(raster)
polygon_spdf <- shapefile(polygon_filepath)

# Specifying output file and path
output <- file.path(outdir, "GEDI01_B_201910080338_001964_T05337_02_003_01_clip")

# clipping GEDI Level1B data by extent boundary box
level1b_clip <- clipLevel1BGeometry(level1b, polygon_spdf = polygon_spdf,
                                      output = output,
                                      split_by = "id")

close(level1b)
lapply(level1b_clip, close)

---

**clipLevel2A**  
*Clip GEDI Level2A data by Coordinates*

**Description**

This function clips GEDI Level2A data within a given bounding coordinates

**Usage**

`clipLevel2A(level2a, xmin, xmax, ymin, ymax, output)`

**Arguments**

- `level2a` A GEDI Level2A object (output of `readLevel2A` function). An S4 object of class "gedi.level2a".
- `xmin` Numeric. West longitude (x) coordinate of the bounding rectangle, in decimal degrees.
- `xmax` Numeric. East longitude (x) coordinate of the bounding rectangle, in decimal degrees.
- `ymin` Numeric. South latitude (y) coordinate of the bounding rectangle, in decimal degrees.
clipLevel2AGeometry

clipLevel2AGeometry
Clip GEDI Level2A data by geometry

$\textbf{ymax}$
Numeric. North latitude (y) coordinate of the bounding rectangle, in decimal degrees.

$\textbf{output}$
Optional character path where to save the new hdf5 file. The default stores a temporary file only.

**Value**

Returns a list of S4 objects of class ”gedi.level2a” containing clipped GEDI Level2A data.

**See Also**

https://lpdaac.usgs.gov/products/gedi02_av001/

**Examples**

```r
outdir = tempdir()

# Specifying the path to GEDI level2A data (zip file)
level2A_fp_zip <- system.file("extdata",
    "GEDI02_A_2019108080338_001964_T05337_02_001_01_sub.zip",
    package="rGEDI")

# Unzipping GEDI level2A data
level2Apath <- unzip(level2A_fp_zip,exdir = outdir)

# Reading GEDI level2A data (h5 file)
level2a<-readLevel2A(level2Apath=level2Apath)

# Bounding rectangle coordinates
xmin=-44.13
xmax=-44.12
ymin=-13.74
ymax=-13.73

# Specifying output file and path
output<-file.path(outdir,"GEDI02_A_2019108080338_001964_T05337_02_001_01_clip.h5")

# Clipping GEDI Level2A data by boundary box extent
level2a_clip <- clipLevel2A(level2a,xmin,xmax,ymin,ymax,output)

close(level2a)
close(level2a_clip)
```
Description

This function clips GEDI Level2A data within a given geometry.

Usage

clipLevel2AGeometry(level2a, polygon_spdf, output="", split_by=NULL)

Arguments

level2a  A GEDI Level2A object (output of readLevel2A function). An S4 object of class "gedi.level2a".
polygon_spdf  Polygon. An object of class SpatialPolygonsDataFrame-class, which can be loaded as an ESRI shapefile using raster::shapefile() function in the raster package.
output  optional character path where to save the new h5file. Default "" (temporary file).
split_by  Polygon id. If defined, GEDI data will be clipped by each polygon using the attribute specified by split_by from the attribute table.

Value

Returns a list of S4 object of class "gedi.level2a" containing clipped GEDI Level2A data.

See Also

https://lpdaac.usgs.gov/products/gedi02_av001/

Examples

outdir = tempdir()

# Specifying the path to GEDI level2A data (zip file)
level2A_fp_zip <- system.file("extdata",
  "GEDI02_A_2019100800338_001964_T05337_02_001_01_sub.zip",
  package="rGEDI")

# Unzipping GEDI level2A data
level2Apath <- unzip(level2A_fp_zip,exdir = outdir)

# Reading GEDI level2A data (h5 file)
level2a<-readLevel2A(level2Apath=level2Apath)

# Specifying the path to shapefile
polygon_filepath <- system.file("extdata", "stands_cerrado.shp", package="rGEDI")

# Reading shapefile as SpatialPolygonsDataFrame object
library(raster)
polygon_spdf<-shapefile(polygon_filepath)

# Specifying output file and path
```r
output <- file.path(outdir, "GEDI02_A_20191008338_001964_T05337_02_001_01_clip")

# Clipping GEDI Level2A data by geometry
level2a_clip <- clipLevel2AGeometry(level2a, polygon_spdf = polygon_spdf,
                                        output = output,
                                        split_by = "id")

close(level2a)
lapply(level2a_clip, close)
```

---

**clipLevel2AM**  
*Clip GEDI Elevation and Height Metrics by Coordinates*

**Description**

This function clips GEDI Level2A extracted Elevation and Height Metrics (Level2AM) within a given bounding coordinates.

**Usage**

```
clipLevel2AM(level2AM, xmin, xmax, ymin, ymax)
```

**Arguments**

- **level2AM**: A GEDI Level2A object (output of `readLevel2A` function). An S4 object of class "gedi.level2a".
- **xmin**: Numeric. West longitude (x) coordinate of bounding rectangle, in decimal degrees.
- **xmax**: Numeric. East longitude (x) coordinate of bounding rectangle, in decimal degrees.
- **ymin**: Numeric. South latitude (y) coordinate of bounding rectangle, in decimal degrees.
- **ymax**: Numeric. North latitude (y) coordinate of bounding rectangle, in decimal degrees.

**Value**

Returns an S4 object of class `data.table-class` containing the clipped elevation and relative heights metrics.

**See Also**

https://lpdaac.usgs.gov/products/gedi02_av001/
Examples

```r
# Specifying the path to GEDI level2A data (zip file)
outdir = tempdir()
level2A_fp_zip <- system.file("extdata", "GEDI02_A_2019108080338_001964_T05337_02_001_01_sub.zip", package="rGEDI")

# Unzipping GEDI level2A data
level2Apath <- unzip(level2A_fp_zip,exdir = outdir)

# Reading GEDI level2A data (h5 file)
level2a<-readLevel2A(level2Apath=level2Apath)

# Extracting GEDI Elevation and Height Metrics
level2AM = getLevel2AM(level2a)

# Bounding rectangle coordinates
xmin = -44.15036
xmax = -44.10066
ymin = -13.75831
ymax = -13.71244

# Clipping GEDI data by boundary box extent
level2AM_clip <- clipLevel2AM(level2AM,xmin, xmax, ymin, ymax)

close(level2a)
```

---

**clipLevel2AMGeometry**  
*Clip GEDI Elevation and Height Metrics by Coordinates*

**Description**

This function clips GEDI Level2A extracted Elevation and Height Metrics (Level2AM) within a given bounding coordinates

**Usage**

```r
clipLevel2AMGeometry(level2AM, polygon_spdf, split_by)
```

**Arguments**

- `level2AM`: A GEDI Level2A object (output of `readLevel2A` function). An S4 object of class "data.table".
- `polygon_spdf`: Polygon. An object of class `SpatialPolygonsDataFrame-class`, which can be loaded as an ESRI shapefile using `raster::shapefile()` function in the `raster` package.
- `split_by`: Polygon id. If defined, GEDI data will be clipped by each polygon using the polygon id from table of attribute defined by the user
Value

Returns an S4 object of class `data.table-class` containing the clipped elevation and relative heights metrics.

Examples

```r
# Specifying the path to GEDI level2A data (zip file)
outdir = tempdir()
level2A_fp_zip <- system.file("extdata",
    "GEDI02_A_20191008000338_001964_T05337_02_001_01_sub.zip",
    package="rGEDI")

# Unzipping GEDI level2A data
level2Apath <- unzip(level2A_fp_zip,exdir = outdir)

# Reading GEDI level2A data (h5 file)
level2a<-readLevel2A(level2Apath=level2Apath)

# Extracting GEDI Elevation and Height Metrics
level2AM = getLevel2AM(level2a)

# Specifying the path to shapefile
polygon_filepath <- system.file("extdata", "stands_cerrado.shp", package="rGEDI")

# Reading shapefile as SpatialPolygonsDataFrame object
library(raster)
polygon_spdf<-shapefile(polygon_filepath)

# Clipping GEDI data by Geometry
level2AM_clip = clipLevel2AMGeometry(level2AM, polygon_spdf, split_by="id")
```

```r
library(leaflet)
leaflet() %>%
  addCircleMarkers(level2AM_clip$lat_lowestmode,
  level2AM_clip$lon_lowestmode,
  radius = 1,
  opacity = 1,
  color = "red") %>%
  addScaleBar(options = list(imperial = FALSE)) %>%
  addPolygons(data=polygon_spdf,weight=1,col = 'white',
  opacity = 1, fillOpacity = 0) %>%
  addProviderTiles(providers$Esri.WorldImagery)
```

```
close(level2a)
```

clipLevel2B  

`Clip GEDI Level2B data by Coordinates`

Description

This function extracts GEDI Level1B data a within given bounding coordinates.
Usage

clipLevel2B(level2b, xmin, xmax, ymin, ymax, output = "")

Arguments

level2b A GEDI Level2B object (output of readLevel2B function). An S4 object of class "gedi.level2b".

xmin Numeric. West longitude (x) coordinate of the bounding rectangle, in decimal degrees.

xmax Numeric. East longitude (x) coordinate of the bounding rectangle, in decimal degrees.

ymin Numeric. South latitude (y) coordinate of the bounding rectangle, in decimal degrees.

ymax Numeric. North latitude (y) coordinate of the bounding rectangle, in decimal degrees.

output Optional character path where to save the new hdf5 file. The default stores a temporary file only.

Value

Returns a list of S4 object of class "gedi.level2b" containing clipped GEDI Level2B data.

See Also

https://lpdaac.usgs.gov/products/gedi01_bv001/

Examples

outdir = tempdir()

# Specifying the path to GEDI level2B data (zip file)
level2B_fp_zip <- system.file("extdata",
    "GEDI02_B_2019108080338_001964_T05337_02_001_01_sub.zip",
    package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)

# Reading GEDI level2B data (h5 file)
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Bounding rectangle coordinates
xmin=-44.13
xmax=-44.12
ymin=-13.74
ymax=-13.73

# Specifying output file and path
clipLevel2BGeometry

Clip GEDI Level2B data by geometry

Description
This function extracts GEDI Level1B data within a given geometry

Usage
clipLevel2BGeometry(level2b, polygon_spdf, output = "", split_by = NULL)

Arguments
level2b A GEDI Level2B object (output of readLevel2B function). An S4 object of class "gedi.level2b".
polygon_spdf Polygon. An object of class SpatialPolygonsDataFrame-class, which can be loaded as an ESRI shapefile using raster::shapefile() function in the raster package.
output optional character path where to save the new h5file. Default "" (temporary file).
split_by Polygon id. If defined, GEDI data will be clipped by each polygon using the attribute specified by split_by from the attribute table.

Value
Returns a list of S4 objects of class "gedi.level2b" containing clipped GEDI Level2B data.

See Also
https://lpdaac.usgs.gov/products/gedi01_bv001/

Examples

outdir = tempdir()

# Specifying the path to GEDI level2B data (zip file)
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_201910080338_001964_T05337_02_001_01_sub.zip",
  package="rGEDI")
# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip, exdir = outdir)

# Reading GEDI level2B data (h5 file)
level2b <- readLevel2B(level2Bpath = level2Bpath)

# Specifying the path to shapefile
polygon_filepath <- system.file("extdata", "stands_cerrado.shp", package = "rGEDI")

# Reading shapefile as SpatialPolygonsDataFrame object
library(raster)
polygon_spdf <- shapefile(polygon_filepath)

# Specifying output file and path
output <- file.path(outdir, "GEDI02_B_201910080338_02_001_01_clip")

# Clipping GEDI data by extent boundary box
level2b_clip <- clipLevel2BGeometry(level2b, polygon_spdf = polygon_spdf,
                                    output = output,
                                    split_by = "id")

close(level2b)
lapply(level2b_clip, close)

---

clipLevel2BPAIProfile  *Clip GEDI Plant Area Index profile by Coordinates*

**Description**
This function clips GEDI level2B derived Plant Area Index profile a within given bounding coordinates

**Usage**

```r
clipLevel2BPAIProfile(level2BPAIProfile, xmin, xmax, ymin, ymax)
```

**Arguments**

- `level2BPAIProfile`  
  A GEDI Level2B object (output of `getLevel2BPAIProfile` function). An S4 object of class "gedi.level2b".

- `xmin`  
  Numeric. West longitude (x) coordinate of the bounding rectangle, in decimal degrees.

- `xmax`  
  Numeric. East longitude (x) coordinate of the bounding rectangle, in decimal degrees.

- `ymin`  
  Numeric. South latitude (y) coordinate of the bounding rectangle, in decimal degrees.

- `ymax`  
  Numeric. North latitude (y) coordinate of the bounding rectangle, in decimal degrees.
clipLevel2BPAIProfileGeometry

ymax

Numeric. North latitude (y) coordinate of the bounding rectangle, in decimal degrees.

Value

Returns an S4 object of class data.table-class containing the Plant Area Index profile data.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv001/

Examples

# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
"GEDI02_B_201910080338_001964_T05337_02_001_01_sub.zip",
package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)

# Reading GEDI level2B data (h5 file)
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Extracting GEDI Plant Area Index profile
level2BPAIProfile<-getLevel2BPAIProfile(level2b)

# Bounding rectangle coordinates
xmin = -44.15036
xmax = -44.10066
ymin = -13.75831
ymax = -13.71244

# Clipping GEDI Plant Area Index profile by extent boundary box
level2b_clip <- clipLevel2BPAIProfile(level2BPAIProfile,xmin, xmax, ymin, ymax)

close(level2b)

clipLevel2BPAIProfileGeometry

Clip GEDI Plant Area Index profile by geometry

Description

This function clips GEDI level2B derived Plant Area Index profile within a given geometry

Usage

clipLevel2BPAIProfileGeometry(level2BPAIProfile, polygon_spdf, split_by)
Arguments

level2BPAIProfile
A GEDI Level2B object (output of `getLevel2BPAIProfile` function). An S4 object of class "data.table".

polygon_spdf
Polygon. An object of class `SpatialPolygonsDataFrame-class`, which can be loaded as an ESRI shapefile using `raster::shapefile()` function in the raster package.

split_by
Polygon id. If defined, GEDI data will be clipped by each polygon using the attribute specified by `split_by` from the attribute table.

Value

Returns an S4 object of class `data.table-class` containing the Plant Area Index profile data.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv001/

Examples

```r
# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata", "GEDI02_B_2019108080338_001964_T05337_02_001_01_sub.zip", package="rGEDI")

# Unzipping GEDI level2B data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)

# Reading GEDI level2B data (h5 file)
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Extracting GEDI Plant Area Index profile
level2BPAIProfile<-getLevel2BPAIProfile(level2b)

# Specifying the path to shapefile
polygon_filepath <- system.file("extdata", "stands_cerrado.shp", package="rGEDI")

# Reading shapefile as SpatialPolygonsDataFrame object
library(raster)
polygon_spdf<-shapefile(polygon_filepath)

# Clipping GEDI Plant Area Index profile by geometry
level2b_clip_geometry <- clipLevel2BPAIProfileGeometry(
  level2BPAIProfile, polygon_spdf, split_by="id")

close(level2b)
```
clipLevel2BPAVDPProfile

Clip GEDI Plant Area Volume Density profile by Coordinates

Description

This function clips GEDI level2B derived Plant Area Volume Density profile within a given bounding coordinates.

Usage

clipLevel2BPAVDPProfile(level2BPAVDPProfile, xmin, xmax, ymin, ymax)

Arguments

level2BPAVDPProfile
A GEDI Level2B object (output of getLevel2BPAVDPProfile function). An S4 object of class "data.table".

xmin
Numeric. West longitude (x) coordinate of the bounding rectangle, in decimal degrees.

xmax
Numeric. East longitude (x) coordinate of the bounding rectangle, in decimal degrees.

ymin
Numeric. South latitude (y) coordinate of the bounding rectangle, in decimal degrees.

ymax
Numeric. North latitude (y) coordinate of the bounding rectangle, in decimal degrees.

Value

Returns an S4 object of class data.table-class containing the Plant Area Volume Density profile data.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv001/

Examples

# specify the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
"GEDI02_B_2019108080338_001964_T05337_02_001_01_sub.zip",
package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)
```r
# Reading GEDI level2B data (h5 file)
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Extracting GEDI Plant Area Volume Density profile
level2BPAVDProfile<-getLevel2BPAVDProfile(level2b)

# Bounding rectangle coordinates
xmin = -44.15036
xmax = -44.10066
ymin = -13.75831
ymax = -13.71244

# Clipping GEDI Plant Area Volume Density profile by boundary box extent
level2BPAVDProfile_clip <- clipLevel2BPAVDProfile(level2BPAVDProfile, xmin, xmax, ymin, ymax)

close(level2b)
```

---

**clipLevel2BPAVDProfileGeometry**

*Clip GEDI Plant Area Volume Density profile by geometry*

**Description**

This function clips GEDI level2B derived Plant Area Index profile within a given geometry.

**Usage**

`clipLevel2BPAVDProfileGeometry(level2BPAVDProfile, polygon_spdf, split_by)`

**Arguments**

- `level2BPAVDProfile`: A GEDI Level2B object (output of `getLevel2BPAIProfile` function). An S4 object of class "gedi.level2b".
- `polygon_spdf`: Polygon. An object of class `SpatialPolygonsDataFrame-class`, which can be loaded as an ESRI shapefile using `raster::shapefile()` function in the `shapefile` package.
- `split_by`: Polygon id. If defined, GEDI data will be clipped by each polygon using the attribute specified by `split_by` from the attribute table.

**Value**

Returns an S4 object of class `data.table-class` containing the Plant Area Volume Density profile data.

**See Also**

`https://lpdaac.usgs.gov/products/gedi02_bv001/`
clipLevel2BVPM

Clip GEDI Canopy Cover and Vertical Profile Metrics by Coordinates

Description

This function clips GEDI level2B derived Canopy Cover and Vertical Profile metrics a within given bounding coordinates

Usage

clipLevel2BVPM(level2BVPM, xmin, xmax, ymin, ymax)

Arguments

level2BVPM A GEDI Level2B object (output of readLevel1B function). An S4 object of class "data.table".

xmin Numeric. West longitude (x) coordinate of the bounding rectangle, in decimal degrees.
clipLevel2BVPM

xmax Numeric. East longitude (x) coordinate of the bounding rectangle, in decimal degrees.

ymin Numeric. South latitude (y) coordinate of the bounding rectangle, in decimal degrees.

ymax Numeric. North latitude (y) coordinate of the bounding rectangle, in decimal degrees.

Value

Returns an S4 object of class **data.table-class** containing the Canopy Cover and Vertical Profile metrics.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv001/

Examples

```r
# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
    "GEDI02_B_2019108080338_001964_T05337_02_001_01_sub.zip",
    package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)

# Reading GEDI level2B data (h5 file)
level2b<readLevel2B(level2Bpath=level2Bpath)

# Extracting canopy cover and vertical profile metrics
level2BVPM<-getLevel2BVPM(level2b)

# Bounding rectangle coordinates
xmin = -44.15036
xmax = -44.10066
ymin = -13.75831
ymax = -13.71244

# Clipping level2BVPM by extent boundary box
level2b_clip <- clipLevel2BVPM(level2BVPM,xmin, xmax, ymin, ymax)

library(leaflet)
leaflet() %>%
  addCircleMarkers(level2b_clip$longitude_bin0, level2b_clip$latitude_bin0, radius = 1, opacity = 1, color = "red") %>%
  addScaleBar(options = list(imperial = FALSE)) %>%
```
### clipLevel2BVPMGeometry

*Clip GEDI Canopy Cover and Vertical Profile Metrics by geometry*

**Description**

This function clips GEDI level2B derived Canopy Cover and Vertical Profile metrics within a given geometry.

**Usage**

```r
clipLevel2BVPMGeometry(level2BVPM, polygon_spdf, split_by)
```

**Arguments**

- `level2BVPM`: A GEDI Level2B object (output of `readLevel1B` function). An S4 object of class "gedi.level2b".
- `polygon_spdf`: Polygon. An object of class `SpatialPolygonsDataFrame-class`, which can be loaded as an ESRI shapefile using `raster::shapefile()` function in the `raster` package.
- `split_by`: Polygon id. If defined, GEDI data will be clipped by each polygon using the attribute specified by `split_by` from the attribute table.

**Value**

Returns an S4 object of class `data.table-class` containing the Canopy Cover and Vertical Profile metrics.

**See Also**

https://lpdaac.usgs.gov/products/gedi02_bv001/

**Examples**

```r
# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
    "GEDI02_B_2019108080338_001964_T05337_02_001_01_sub.zip",
package="rGEDI")

# Unzipping GEDI level2B data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)
```
# Reading GEDI level2B data (h5 file)
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Extracting canopy cover and vertical profile metrics
level2BVPM<-getLevel2BVPM(level2b)

# Specifying the path to shapefile
polygon_filepath <- system.file("extdata", "stands_cerrado.shp", package="rGEDI")

# Reading shapefile as SpatialPolygonsDataFrame object
library(raster)
polygon_spdf<-shapefile(polygon_filepath)

# Clipping level2BVPM by geometry
level2b_clip_geometry <- clipLevel2BVPMGeometry(level2BVPM,polygon_spdf,split_by="id")

library(leaflet)
leaflet() %>%
  addCircleMarkers(level2b_clip_geometry$longitude_bin0,
                   level2b_clip_geometry$latitude_bin0,
                   radius = 1,
                   opacity = 1,
                   color = "red") %>%
  addScaleBar(options = list(imperial = FALSE)) %>%
  addPolygons(data=polygon_spdf,weight=1,col = "white",
              opacity = 1, fillOpacity = 0) %>%
  addProviderTiles(providers$Esri.WorldImagery)

close(level2b)

---

**close**  
*Close hdf5 connections from gedi* objects

**Description**

Closing files will avoid locking HDF5 GEDI files.

**Usage**

```r
close(con, ...)
```

### S4 method for signature 'gedi.level1b'
```r
close(con, ...)
```

### S4 method for signature 'gedi.level2a'
```r
close(con, ...)
```
## S4 method for signature 'gedi.level2b'

`close(con, ...)`

## S4 method for signature 'gedi.level1bSim'

`close(con, ...)`

### Arguments

- **con**: An object of class `gedi`
- **...**: Inherited from base

---

### gedi.fullwaveform-class

*Class for GEDI level1B Full Waveform*

### Description

Class for GEDI level1B Full Waveform

### Slots

- **dt**: Object of class `data.table` from `data.table` package containing the extracted GEDI full-waveform elevation and amplitude.

---

### gedi.level1b-class

*Class for GEDI level1B*

### Description

Class for GEDI level1B

### Slots

- **h5**: Object of class H5File from `hdf5r` package containing the GEDI level1B products: geolocated Waveforms

### See Also

- `hdf5r::H5File` in the `hdf5r` package and https://lpdaac.usgs.gov/products/gedi01_bv001/
**gedi.level1bSim-class**  
*Class for GEDI Full-Waveform Simulation*

**Description**

Class for GEDI Full-Waveform Simulation

**Slots**

- `h5` Object of class H5File from `hdf5r` package containing the simulated GEDI full-waveform

**See Also**


ii) gediSimulator: https://bitbucket.org/StevenHancock/gedisimulator/src/master/

---

**gedi.level2a-class**  
*Class for GEDI level2A*

**Description**

Class for GEDI level2A

**Slots**

- `h5` Object of class H5File from `hdf5r` package containing the GEDI level2A products: ground elevation, canopy top height, and relative heights (RH).

**See Also**

- `hdf5r::H5File` in the `hdf5r` package and https://lpdaac.usgs.gov/products/gedi02_av001/
**gedi.level2b-class**  
*Class for GEDI level2B*

**Description**

Class for GEDI level2B

**Slots**

- `h5` Object of class H5File from `hdf5r` package containing the GEDI level2B products: canopy cover, Plant Area Index (PAI), Plant Area Volume Density (PAVD), and Foliage Height Diversity (FHD).

**See Also**

- `hdf5r::H5File` in the `hdf5r` package and https://lpdaac.usgs.gov/products/gedi02_bv001/

---

**gediDownload**  
*Download GEDI data*

**Description**

Download GEDI data from LP DAAC Data Pool. Users will need to enter their Earth Explore login Information for downloading the data.

**Usage**

```r
gediDownload(filepath, outdir = NULL, overwrite = FALSE, buffer_size = 512)
```

**Arguments**

- `filepath` Vector object; path to the GEDI data
- `outdir` Vector object, output directory for downloading GEDI data, default `tempdir()`
- `overwrite` logical; overwrite file if they already exists in destination, default FALSE
- `buffer_size` integer; the size of download chunk in KB to hold in memory before writing to file, default 512.

**Value**

No return value on success, on failure it will `stop()`

**References**

## Examples

```r
# Set path to GEDI data
# herein we will only download xml metadata
filepath=c(paste0(
    "https://e4ftl01.cr.usgs.gov/GEDI/GEDI02_B.001",
    "/2019.04.18/GEDI02_B_2019108032534_O01961_T03911_02_001_01.h5.xml"
 ),
    paste0("https://e4ftl01.cr.usgs.gov/GEDI/GEDI02_B.001",
    "/2019.04.18/GEDI02_B_2019108045815_O01962_T01066_02_001_01.h5.xml"
 ))

# Set dir to download files to
outdir=tempdir()

# Downloading GEDI data
gediDownload(filepath, outdir)
```

### gedifinder

**GEDI finder**

## Description

This function finds the exact granule(s) that contain GEDI data for a given region of interest and date range.

## Usage

```r
gediofinder(
  product,
  ul_lat,
  ul_lon,
  lr_lat,
  lr_lon,
  version = "001",
  daterange = NULL
)
```

## Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>product</td>
<td>GEDI data level; Options: &quot;GEDI01_B&quot;, &quot;GEDI02_A&quot; or &quot;GEDI02_B&quot;</td>
</tr>
<tr>
<td>ul_lat</td>
<td>Numeric. Upper left (ul) corner coordinates, in lat (decimal degrees) for the bounding box of the area of interest.</td>
</tr>
<tr>
<td>ul_lon</td>
<td>Numeric. Upper left (ul) corner coordinates, in lon (decimal degrees) for the bounding box of the area of interest.</td>
</tr>
</tbody>
</table>
lr_lat Numeric. Lower right (ul) corner coordinates, in lat (decimal degrees) for the bounding box of the area of interest.

lr_lon Numeric. Lower right (ul) corner coordinates, in lon (decimal degrees) for the bounding box of the area of interest.

version Character. The version of the GEDI product files to be returned. Default "001".

daterange Vector. Date range. Specify your start and end dates (year-month-day). Ex.: c("2019-07-01","2020-05-22"). If NULL (default), the date range filter will be not applied.

Value

Return a vector object pointing out the path saving the downloaded GEDI data within the boundary box coordinates provided

See Also

bbox: Defined by the upper left and lower right corner coordinates, in lat,lon ordering, for the bounding box of the area of interest (e.g. [ul_lat,ul_lon,lr_lat,lr_lon]). This function relies on the existing LP DAAC gedifinder tool: https://lpdaacsfc.cr.usgs.gov/services/gedifinder

Examples

# gedifinder is a web service provided by NASA
# usually the request takes more than 5 seconds

# Specifying bounding box coordinates
ul_lat<- 42.0
ul_lon<- -100
lr_lat<- 40.0
lr_lon<- -96.0

# Specifying the date range
daterange=c("2019-07-01","2020-05-22")

# Extracting the path to GEDI data for the specified boundary box coordinates
gedi02b_list<-gedifinder(product="GEDI02_B",
ul_lat,
ul_lon,
lr_lat,
lr_lon,
version="001",
daterange=daterange)
Description

GEDI full waveform data processing and metrics extraction

Usage

gediWFMetrics(
    input,
    outRoot,
    writeFit = FALSE,
    writeGauss = FALSE,
    bounds = NULL,
    ground = FALSE,
    useInt = FALSE,
    useFrac = FALSE,
    rhRes = 5,
    laiRes = 10,
    laiH = 30,
    noRHgauss = FALSE,
    gTol = 0,
    fhdHistRes = 0.001,
    forcePsigma = FALSE,
    bayesGround = FALSE,
    dontTrustGround = FALSE,
    noRoundCoord = FALSE,
    noCanopy = FALSE,
    dcBias = 0,
    nSig = 0,
    hNoise = 0,
    linkNoise = NULL,
    linkFsig = NULL,
    linkPsig = NULL,
    trueSig = NULL,
    bitRate = NULL,
    maxDN = NULL,
    renoise = FALSE,
    newPsig = -1,
    oldPsig = 0.764331,
    addDrift = NULL,
    missGround = FALSE,
    minGap = NULL,
    photonCount = FALSE,
    pcl = FALSE,
    nPhotons = 2.1,
gediWFMetrics

    photonWind = 200,
noiseMult = 0.1,
rhoVrhoG = 1,
nPhotC = 2.1,
nPhotG = -1,
photHDF = FALSE,
meanN = 0,
thresh = 1e-14,
varNoise = FALSE,
varScale = NULL,
statsLen = NULL,
noiseTrack = FALSE,
sWidth = NULL,
psWidth = 0,
msWidth = NULL,
preMatchF = FALSE,
postMatchF = FALSE,
pFile = NULL,
gWidth = 1.2,
minGsigt = 0.764331,
minWidth = 0,
medNoise = FALSE,
varDrift = NULL,
DriftFac = NULL,
rhoG = 0.4,
rhoC = 0.57,
pSigma = NULL,
gold = FALSE,
deconTol = NULL
)

Arguments

input  
      gedi.level1bSim (may be a list of objects). Simulated waveform input object(s).
outRoot  name. output filename root
writeFit  write fitted waveform
writeGauss  write Gaussian parameters
bounds  minX minY maxX maxY. only analyse data within bounds
ground  read true ground from file
useInt  use discrete intensity instead of count
useFrac  use fractional hits rather than counts
rhRes  r. percentage energy resolution of RH metrics
laiRes  res. lai profile resolution in metres
laiH  h. height to calculate LAI to
noRHgauss  do not fit Gaussians
gtol  tol. ALS ground tolerance. Used to calculate slope.
fhdHistRes  res. waveform intensity resolution to use when calculating FHD from histograms
forcePsigma  do not read pulse sigma from file
bayesGround  use Bayesian ground finding
dontTrustGround  don’t trust ground in waveforms, if included
noRoundCoord  do not round up coords when outputting
noCanopy  do not calculate FHD histograms and LAI profiles
dcBias  n. mean noise level
nSig  sig. noise sigma
hNoise  n. hard threshold noise as a fraction of integral
linkNoise  linkM cov. apply Gaussian noise based on link margin at a cover
linkFsig  sig. footprint width to use when calculating and applying signal noise
linkPsig  sig. pulse width to use when calculating and applying signal noise
trueSig  sig. true sigma of background noise
bitRate  n. digitisation bit rate
maxDN  max. maximum DN
renoise  remove noise from truth before applying new noise level
newPsig  sig. new value for pulse width, when lengthening pulse
oldPsig  sig. old value for pulse width if not defined in waveform file, when lengthening pulse
addDrift  xi. apply detector background drift
missGround  assume ground is missed to assess RH metrics
minGap  gap. delete signal beneath min detectable gap fraction
photonCount  output point cloud from photon counting
pcl  convert to photon counting pulsecompressed
nPhotons  n. mean number of photons
photonWind  x. window length for photon counting search, metres
noiseMult  x. noise multiplier for photoncounting
rhoVrhoG  x. ratio of canopy to ground reflectance at this wavelength. Not different from rhoV and rhoG
nPhotC  n. mean number of canopy photons (replaces nPhotons and rhoVrhoG)
nPhotG  n. mean number of ground photons (replaces nPhotons and rhoVrhoG)
photHDF  write photoncounting
meanN  n. mean noise level, if using a predefined mean level
thresh  n. noise threshold, if using a predefined noise threshold
varNoise  use a variable noise threshold
varScale  x. variable noise threshold scale (multiple of stdev above mean to set threshold)  
statsLen  len. length to calculate noise stats over for varNoise  
noiseTrack  use noise tracking  
sWidth  sig. smoothing width, after denoising  
psWidth  sigma. smoothing width, before denoising  
msWidth  sig. smoothing width, after noise stats, before denoising  
preMatchF  matched filter before denoising  
postMatchF  matched filter after denoising  
pFile  file. read pulse file, for deconvoluction and matched filters  
gWidth  sig. Gaussian parameter selection smoothing width  
minGsig  sig. minimum Gaussian sigma to fit  
minWidth  n. minimum feature width in bins  
medNoise  use median stats rather than mean  
varDrift  correct detector drift with variable factor  
driftFac  xi. fix drift with constant drift factor  
rhoG  rho. ground reflectance  
rhoC  rho. canopy reflectance  
pSigma  sig. pulse width to smooth by if using Gaussian pulse  
gold  deconvolve with Gold’s method  
deconTol  deconvolution tolerance

Details

a) Metrics descriptions

a.1) Metrics available to GEDI

• **gHeight** Ground elevation (m) from Gaussian fitting  
• **maxGround** Ground elevation (m) from lowest maximum  
• **inflGround** Ground elevation (m) from inflection points.  
• **signal top** Elevation of first point above noise (may include noise tracking).  
• **signal bottom** Elevation of last return above noise (may include noise tracking).  
• **cover** Canopy cover (fraction) from area of Gaussian fitted ground. Uses rho_v=0.57 and rho_g=0.4.  
• **leading edge ext** Leading edge extent (m), from Lefksy et al (2007).  
• **trailing edge extent** Trailing edge extent (m), from Lefksy et al (2007).  
• **rhGauss 0-100** RH metrics, 0  
• **rhMax 0-100** RH metrics, 0  
• **rhInfl 0-100** RH metrics, 0  
• **gaussHalfCov** Canopy cover (fraction) from double the energy beneath the Gaussian ground. Uses rho_v=0.57 and rho_g=0.4.
• *maxHalfCov* Canopy cover (fraction) from double the energy beneath the lowest maximum ground. Uses \( \rho_v=0.57 \) and \( \rho_g=0.4 \).

• *infHalfCov* Canopy cover (fraction) from double the energy beneath the inflection point ground. Uses \( \rho_v=0.57 \) and \( \rho_g=0.4 \).

• *bayHalfCov* Canopy cover (fraction) from double the energy beneath the experimental "Bayesian" ground. Uses \( \rho_v=0.57 \) and \( \rho_g=0.4 \).

• *lon* Footprint centre longitude in projection of ALS data (m).

• *lat* Footprint centre latitude in projection of ALS data (m).

• *waveEnergy* Total energy within waveform (will be 1 scaled by noise for simulations).

• *blairSense* Blair’s sensitivity metric. Canopy cover at which this SNR would have 90

• *FHD* Foliage height diversity

• *niM2* Wenge Ni’s biomass metric, equal to the sum of the RH metrics to the power of 2 (unpublished)

• *niM2.1* Wenge Ni’s biomass metric, equal to the sum of the RH metrics to the power of 2.1 (unpublished)

a.2) Metrics unavailable to GEDI

• *wave ID* Waveform label, relates to plot name and footprint number.

• *true ground* Ground elevation (m) from ALS. Centre of gravity of ground points within footprint

• *true top* Levation of highest point of waveform (m), without noise. Includes pulse blurring.

• *ground slope* Effective ground slope (degrees), from width of ground return. Includes roughness.

• *ALS cover* Canopy cover (fraction) from ALS data. Uses \( \rho_v=0.57 \) and \( \rho_g=0.4 \).

• *rhReal 0-100* RH metrics, 0

• *groundOverlap* Fraction of ground return overlapping with canopy return. A measure of understorey.

• *groundMin* Depth of minimum between ground and canopy return. A measure of understorey.

• *groundInfl* \( \frac{d^2y}{dx^2} \) of inflection point between ground and canopy return. A measure of understorey.

• *pointDense* Average ALS point density within GEDI footprint.

• *beamDense* Average ALS beam density within GEDI footprint.

a.3) System settings

• *pSigma* GEDI system pulse width, sigma (m).

• *fSigma* GEDI footprint width, sigma (m).

• *linkM* Link margin if noise is added (db).

• *linkCov* Canopy cover at which the above link margin is true (fraction).

• *filename* Name of input waveform file.

b) Signal processing description
• **Gaussian fitting** Used for "gHeight", "rhGauss" and "gaussHalfCov". The waveform is denoised (mean+5*sigma, noise tracking to avoid truncation), smoothed (pSigma*0.75) and Gaussians fitted with Levenberg-Marquardt optimisation. The center of the lowest Gaussian containing at least 0.5

• **Maximum** Used for "maxGround", "rhMax" and "maxHalfCov". The waveform is denoised (mean+5*sigma, noise tracking to avoid truncation), smoothed (pSigma*0.75). The lowest maximum is taken as the ground.

• **Inflection points** Used for "inflGround", "rhInfl" and "inflHalfCov". The waveform is denoised (mean+5*sigma, noise tracking to avoid truncation), smoothed (pSigma*0.75). The centre of gravity between the lowest two inflection points is taken as the ground.

• **Half covers** Used for "gaussHalfCov", "maxHalfCov" and "inflHalfCov". Sum energy beneath estimated ground position. Double that is the ground energy. Calculate canopy cover, correcting for rho_v and rho_g.

\[
\text{cover} = \frac{E_{can}}{E_{can} + E_{g} \cdot \rho_{v}^2 \rho_{g}}
\]

Where Ecan is the canopy energy, Eg is the ground energy, rho_v is the vegetation reflectance and rho_g is the ground reflectance.


**Value**

Returns a list of metrics derived from the simulated full waveform. A text file (txt) containing the metrics will be saved in the output folder (outRoot). Please see the details section for checking the definition of the metrics.

**See Also**


ii) gediSimulator: https://bitbucket.org/StevenHancock/gedisimulator/src/master/

**Examples**

```r
definePath() if (!definedPath) {
 # Specifying the path to ALS data (zip)
 alsfile_Amazon_zip <- system.file("extdata", "Amazon.zip", package="rGEDI")
 alsfile_Savanna_zip <- system.file("extdata", "Savanna.zip", package="rGEDI")

 # Unzipping ALS data
 alsfile_Amazon_filepath <- unzip(alsfile_Amazon_zip, exdir = definedPath)
alsfile_Savanna_filepath <- unzip(alsfile_Savanna_zip, exdir = outdir)

# Reading and plot ALS file (las file)
als_Amazon <- readLAS(alsfile_Amazon_filepath)
als_Savanna <- readLAS(alsfile_Savanna_filepath)

# Extracting plot center geolocations
xcenter_Amazon = mean(als_Amazon@bbox[,1])
ycenter_Amazon = mean(als_Amazon@bbox[,2])
xcenter_Savanna = mean(als_Savanna@bbox[,1])
ycenter_Savanna = mean(als_Savanna@bbox[,2])

# Simulating GEDI full waveform
wf_Amazon <- gediWFSimulator(input=alsfile_Amazon_filepath,
                             output=file.path(outdir, "gediWF_amazon_simulation.h5"),
                             coords = c(xcenter_Amazon, ycenter_Amazon))

wf_Savanna <- gediWFSimulator(input=alsfile_Savanna_filepath,
                               output=file.path(outdir, "gediWF_Savanna_simulation.h5"),
                               coords = c(xcenter_Savanna, ycenter_Savanna))

# Extracting GEDI full waveform derived metrics without adding noise to the full waveform
wf_amazon_metrics <- gediWFMetrics(input=wf_Amazon, outRoot=file.path(outdir, "amazon"))
wf_savanna_metrics <- gediWFMetrics(input=wf_Savanna, outRoot=file.path(outdir, "savanna"))

metrics <- rbind(wf_amazon_metrics, wf_savanna_metrics)
rownames(metrics) <- c("Amazon", "Savanna")
head(metrics)

# Extracting GEDI full waveform derived metrics after adding noise to the waveform
wf_amazon_metrics_noise <- gediWFMetrics(input=wf_Amazon,
                                         outRoot=file.path(outdir, "amazon"),
                                         linkNoise = c(3.0103, 0.95),
                                         maxDN = 4096,
                                         sWidth = 0.5,
                                         varScale = 3)

wf_savanna_metrics_noise <- gediWFMetrics(
                                       input=wf_Savanna,
                                       outRoot=file.path(outdir, "savanna"),
                                       linkNoise = c(3.0103, 0.95),
                                       maxDN = 4096,
                                       sWidth = 0.5,
                                       varScale = 3)

close(wf_Amazon)
close(wf_Savanna)

metrics_noise <- rbind(wf_amazon_metrics_noise, wf_savanna_metrics_noise)
rownames(metrics_noise) <- c("Amazon", "Savanna")
head(metrics_noise)
gediWFSimulator

**Description**

Simulate GEDI full waveform data from Airborne Laser Scanning (ALS) 3D point cloud

Input and output filenames, and formats

**Usage**

```r
gediWFSimulator(
  input,
  output,
  waveID = NULL,
  coords = NULL,
  listCoord = NULL,
  gridBound = NULL,
  gridStep = 30,
  pSigma = -1,
  pFWHM = 15,
  readPulse = NULL,
  fSigma = 5.5,
  wavefront = NULL,
  res = 0.15,
  topHat = FALSE,
  sideLobe = FALSE,
  lobeAng = 0,
  checkCover = FALSE,
  maxScanAng = 1e+06,
  decimate = 1,
  pBuff = as.integer(2e+08),
  maxBins = as.integer(1024),
  countOnly = FALSE,
  pulseAfter = FALSE,
  pulseBefore = TRUE,
  noNorm = FALSE,
  noOctree = FALSE,
  octLevels = as.integer(0),
  nOctPix = as.integer(40),
  keepOld = FALSE,
  useShadow = FALSE,
  polyGround = FALSE
)
```

**Arguments**

- **input** character vector. lasfile input filename
output character. output filename
waveID id. supply a waveID to pass to the output (only for single footprints) Single footprint, list of footprints, or grid of footprints
coords lon lat numeric vector. footprint coordinate in same system as lasfile
listCoord name. Text file with list of coordinates. Pattern: X Y ['waveID'] ['geoCoordsX'] ['geoCoordsY']. '[]' are optional, separated by spaces.
gridBound minX maxX minY maxY numeric vector. make a grid of waveforms in this box
gridStep res. grid step size Lidar characteristics. Defaults are expected GEDI values.
pSigma pSigmasig. set Gaussian pulse width as 1 sigma
pFWHM fhwm. set Gaussian pulse width as FWHM in ns
readPulse file. read pulse shape and width from a file instead of making Gaussian
fSigma sig. set footprint width
wavefront file. read wavefront shape from file instead of setting Gaussian. Note that footprint width is still set by fSigma
res res. range resolution of waveform digitisation to output, in units of ALS data
topHat use a top hat wavefront
sideLobe use side lobes
lobeAng ang. lobe axis azimuth Input data quality filters
checkCover check that the footprint is covered by ALS data. Do not output if not
maxScanAng ang. maximum scan angle, degrees
decimate x. probability of accepting an ALS beam Computational speed options
pBuff s. point reading buffer size in Gbytes
maxBins for HDF5, limiting number of bins to save trimming.
countOnly only use count method
pulseAfter apply the pulse smoothing after binning for computational speed, at the risk of aliasing (default)
pulseBefore apply the pulse smoothing before binning to avoid the risk of aliasing, at the expense of computational speed
noNorm don’t normalise for ALS density Octree
noOctree do not use an octree
octLevels n. number of octree levels to use
nOctPix n. number of octree pixels along a side for the top level Using full waveform input data (not tested)
keepOld do not overwrite old files, if they exist
useShadow account for shadowing in discrete return data through voxelization
polyGround find mean ground elevation and slope through fitting a polynomial
#
Value

Returns an S4 object of class `hdf5r::H5File` form the `hdf5r` package containing the simulated GEDI full-waveform.

See Also


ii) gediSimulator: https://bitbucket.org/StevenHancock/gedisimulator/src/master/

Examples

```r
libsAvailable = require(lidR) && require(plot3D)
if (libsAvailable) {
  outdir=tempdir()

  # specify the path to ALS data (zip)
  alsfile_Amazon_zip <- system.file("extdata", "Amazon.zip", package="rGEDI")
  alsfile_Savanna_zip <- system.file("extdata", "Savanna.zip", package="rGEDI")

  # Unzipping ALS data
  alsfile_Amazon_filepath <- unzip(alsfile_Amazon_zip,exdir = outdir)
  alsfile_Savanna_filepath <- unzip(alsfile_Savanna_zip,exdir = outdir)

  # Reading and plot ALS file (las file)
  als_Amazon<-readLAS(alsfile_Amazon_filepath)
  als_Savanna<-readLAS(alsfile_Savanna_filepath)

  # Extracting plot center geolocations
  xcenter_Amazon = mean(als_Amazon@bbox[1,])
  ycenter_Amazon = mean(als_Amazon@bbox[2,])
  xcenter_Savanna = mean(als_Savanna@bbox[1,])
  ycenter_Savanna = mean(als_Savanna@bbox[2,])

  # Simulating GEDI full waveform
  wf_Amazon<-gediWFSimulator(input=alsfile_Amazon_filepath,
                              output=file.path(outdir,"gediWF_amazon_simulation.h5"),
                              coords = c(xcenter_Amazon, ycenter_Amazon))

  wf_Savanna<-gediWFSimulator(input=alsfile_Savanna_filepath,
                               output=file.path(outdir,"gediWF_Savanna_simulation.h5"),
                               coords = c(xcenter_Savanna, ycenter_Savanna))

  # Plot ALS and GEDI simulated full waveform
  oldpar<-par()
  par(mfrow=c(2,2), mar=c(4,4,0,0), oma=c(0,0,1,1),cex.axis = 1.2)
  scatter3D(
    als_Amazon@data$X, als_Amazon@data$Y, als_Amazon@data$Z,
    ALSfile_Amazon,
    xlab="X", ylab="Y", zlab="Z",
    main="ALS Amazon")

  scatter3D(
    als_Amazon@data$X, als_Amazon@data$Y, als_Amazon@data$Z,
    wf_Amazon$hdf5file, 
    xlab="X", ylab="Y", zlab="Z",
    main="GEDI Amazon")

  scatter3D(
    als_Savanna@data$X, als_Savanna@data$Y, als_Savanna@data$Z,
    ALSfile_Savanna,
    xlab="X", ylab="Y", zlab="Z",
    main="ALS Savanna")

  scatter3D(
    als_Savanna@data$X, als_Savanna@data$Y, als_Savanna@data$Z,
    wf_Savanna$hdf5file, 
    xlab="X", ylab="Y", zlab="Z",
    main="GEDI Savanna")
}
```
Get GEDI Full Waveform Geolocations (GEDI Level1B)

Description

This function extracts Pulse Full Waveform Geolocations from GEDI Level1B data

Usage

getLevel1BGeo(level1b, select)

Arguments

level1b  A GEDI Level1B object (output of readLevel1B function). An S4 object of class "gedi.level1b".

select  A character vector specifying the fields to extract from GEDI Level1B data. If NULL, by default it will extract latitude_bin0, latitude_lastbin, longitude_bin0, longitude_lastbin, and shot_number. See details for more options.
Details

Besides of the fields specified as default, GEDI mission level 1B description (https://lpdaac.usgs.gov/products/gedi01_bv001/) details all the information that can be extracted:

- **all_samples_sum** Sum of all values within the 10 km range window.
- **beam** Beam number.
- **channel** Channel number.
- **master_frac** Master time, fractional part.
- **master_int** Master time, integer part.
- **noise_mean_corrected** Noise mean.
- **noise_stddev_corrected** Corrected noise standard deviation.
- **nsemean_even** Noise mean of the beam’s detector channel from even sub-converter.
- **nsemean_odd** Noise mean of the beam’s odd sub-converter.
- **rx_energy** Integrated energy in receive (RX) waveform after subtracting the noise mean.
- **rx_offset** Time interval from first stored sample to first downloaded RX sample.
- **rx_open** Time interval from time 0 to first stored RX sample.
- **rx_sample_count** The number of sample intervals (elements) in each RX waveform.
- **rx_sample_start_index** The index in the rxwaveform dataset of the first element of each RX waveform starting at 1.
- **selection_stretchers_x** Commanded number of samples added to the algorithm section on the left.
- **selection_stretchers_y** Commanded number of samples added to the algorithm section on the right.
- **shot_number** Unique shot identifier.
- **stale_return_flag** Indicates that a "stale" cue point from the coarse search algorithm is being used.
- **th_left_used** Count values for the left threshold used in fine search where two consecutive points at or above this value indicate pulse detection.
- **tx_egamplitude** Amplitude of the extended Gaussian fit to the transmit (TX) waveform.
- **tx_egamplitude_error** Error on tx_egamplitude.
- **tx_egbias** Bias of the extended Gaussian fit to the TX waveform.
- **tx_egbias_error** Error on tx_egbias.
- **tx_egflag** Extended Gaussian fit status flag.
- **tx_eggamma** Gamma value of the extended Gaussian fit to the TX waveform.
- **tx_eggamma_error** Error on tx_eggamma.
- **tx_egsigma** Sigma of the extended Gaussian fit to the TX waveform.
- **tx_egsigma_error** Error on tx_egsigma.
- **tx_gloc** Location (mean) of the Gaussian fit to the TX waveform.
- **tx_gloc_error** Error on tx_gloc.
- **tx_pulsedflag**: Set to 1 if a pulse is detected in the TX waveform.
- **tx_sample_count**: The number of sample intervals (elements) in each transmit waveform.
- **tx_sample_start_index**: The index in the rxwaveform dataset of the first element of each RX waveform starting at 1.
- **altitude_instrument**: Height of the instrument diffractive optical element (DOE) above the WGS84 ellipsoid.
- **altitude_instrument_error**: Error on altitude_instrument.
- **bounce_time_offset_bin0**: The difference between the TX time and the time at the start of the RX window.
- **bounce_time_offset_bin0_error**: Error on bounce_time_offset_bin0.
- **bounce_time_offset_lastbin**: The difference between the TX time and the time at the end of the RX window.
- **bounce_time_offset_lastbin_error**: Error on bounce_time_offset_lastbin.
- **degrade**: Greater than zero if the shot occurs during a degrade period, zero otherwise.
- **delta_time**: Transmit time of the shot, measured in seconds since 2018-01-01.
- **digital_elevation_model**: Digital elevation model height above the WGS84 ellipsoid.
- **elevation_bin0**: Height of the start of the RX window, relative to the WGS-84 ellipsoid.
- **elevation_bin0_error**: Error on elevation_bin0.
- **elevation_lastbin**: Height of the end of the RX window, relative to the WGS-84 ellipsoid.
- **elevation_lastbin_error**: Error on elevation_lastbin.
- **latitude_bin0**: Latitude of the start of the RX window.
- **latitude_bin0_error**: Error on latitude_bin0.
- **latitude_lastbin**: Latitude of the end of the RX window.
- **latitude_lastbin_error**: Error on latitude_lastbin.
- **latitude_instrument**: Latitude of the instrument diffractive optical element (DOE) at laser transmit time.
- **latitude_instrument_error**: Error on latitude_instrument.
- **local_beam_azimuth**: Azimuth of the unit pointing vector for the laser in the local East, North, Up (ENU) frame.
- **local_beam_azimuth_error**: Error on local_beam_azimuth.
- **local_beam_elevation**: Elevation of the unit pointing vector for the laser in the local ENU frame.
- **local_beam_elevation_error**: Error on local_beam_elevation.
- **longitude_bin0**: Longitude of the start of the RX window.
- **longitude_bin0_error**: Error on longitude_bin0.
- **longitude_lastbin**: Longitude of the end of the RX window.
- **longitude_lastbin_error**: Error on longitude_lastbin.
- **longitude_instrument**: Longitude of the instrument diffractive optical element (DOE) at laser transmit time.
- `longitude_instrument_error` Error on longitude_instrument.
- `mean_sea_surface` Mean sea surface height above the WGS84 ellipsoid, includes the geoid.
- `neutat_delay_derivative_bin0` Change in neutral atmospheric delay per height change for the start of the RX window.
- `neutat_delay_derivative_lastbin` Change in neutral atmospheric delay per height change for the end of the RX window.
- `neutat_delay_total_bin0` Total neutral atmosphere delay correction (wet+dry) from the TX pulse to the start of the RX window.
- `neutat_delay_total_lastbin` Total neutral atmosphere delay correction (wet+dry) from the TX pulse to the end of the RX window.
- `range_bias_correction` The range bias applied to the range measurement.
- `shot_number` Unique shot identifier Number.
- `solar_azimuth` The azimuth of the sun position vector.
- `solar_elevation` The elevation of the sun position vector.
- `surface_type` Flags describing which surface types.
- `delta_time` Transmit time of the shot.
- `dynamic_atmosphere_correction` Dynamic Atmospheric Correction (DAC) includes inverted barometer (IB) effect.
- `geoid` Geoid height above WGS-84 reference ellipsoid.
- `tide_earth` Solid Earth tides.
- `tide_load` Load Tide - Local displacement due to Ocean Loading.
- `tide_ocean` Ocean Tides including diurnal and semi-diurnal, and longer period tides.
- `tide_ocean_pole` Oceanic surface rotational deformation due to polar motion.
- `tide_pole` Solid Earth Pole Tide. Rotational deformation due to polar motion.

Value

Returns an S4 object of class `data.table-class` containing the GEDI Full Waveform Geolocations

See Also

https://lpdaac.usgs.gov/products/gedi01_bv001/

Examples

```r
# specify the path to GEDI level1B data (zip file)
outdir = tempdir()
level1B_fp_zip <- system.file("extdata",
"GEDI01_B_2019108080338_001964_T05337_02_003_01_sub.zip",
package="rGEDI")

# Unzipping GEDI level1B data
level1Bpath <- unzip(level1B_fp_zip,exdir = outdir)
```
getLevel1BWF

# Reading GEDI level1B data (h5 file)
level1b <- readLevel1B(level1Bpath = level1Bpath)

# Extracting GEDI level1B geolocations
level1bGeo <- getLevel1BGeo(level1b, select = c("elevation_bin0", "elevation_lastbin"))
head(level1bGeo)
close(level1b)

---

getLevel1BWF

Get GEDI Pulse Full Waveform (GEDI Level1B)

Description

This function extracts the full waveform of a given pulse from GEDI Level1B data.

Usage

getLevel1BWF(level1b, shot_number)

Arguments

level1b  A GEDI Level1B object (output of readLevel1B function). A S4 object of class "gedi.level1b".
shot_number  Shot number. A scalar representing the shot number of a giving pulse.

Details

Shot numbers can be extracted using readLevel1B function.

Value

Returns an S4 object of class "gedi.fullwaveform".

See Also

https://lpdaac.usgs.gov/products/gedi01_bv001/

Examples

# Specifying the path to GEDI level1B data (zip file)
outdir <- tempdir()
level1B_fp_zip <- system.file("extdata", "GEDI01_B_2019108080338_001964_T05337_02_003_01_sub.zip", package="rGEDI")

# Unzipping GEDI level1B data
level1Bpath <- unzip(level1B_fp_zip, exdir = outdir)
# Reading GEDI level1B data (h5 file)
level1b <- readLevel1B(level1Bpath = level1Bpath)

# Extracting GEDI full waveform for a giving shotnumber
wf <- getLevel1BWF(level1b, shot_number = "19640521100108408")

# Plotting GEDI Full waveform
oldpar <- par()
par(mfrow = c(1,2), cex.axis = 1.5)
plot(wf, relative = FALSE, polygon = TRUE, type = "l", lwd = 2, col = "forestgreen",
     xlab = "Waveform Amplitude", ylab = "Elevation (m)")

plot(wf, relative = TRUE, polygon = TRUE, type = "l", lwd = 2, col = "forestgreen",
     xlab = "Waveform Amplitude (%)", ylab = "Elevation (m)")

par(oldpar)
close(level1b)

---

getLevel2AM | Get GEDI Elevation and Height Metrics (GEDI Level2A)

Description

This function extracts Elevation and Relative Height (RH) metrics from GEDI Level2A data.

Usage

getLevel2AM(level2a)

Arguments

level2a | A GEDI Level2A object (output of readLevel2A function). An S4 object of class "gedi.level2a".

Value

Returns an S4 object of class data.table-class containing the elevation and relative heights.

See Also

https://lpdaac.usgs.gov/products/gedi02_av001/

Examples

# Specifying the path to GEDI level2A data (zip file)
outdir = tempdir()
level2A_fp_zip <- system.file("extdata",
                           "GEDI02_A_2019108080338_001964_T05337_02_001_01_sub.zip",
                           package = "rGEDI")
getLevel2BPAIProfile

Get GEDI Plant Area Index (PAI) Profile (GEDI Level2B)

description
This function extracts the Plant Area Index (PAI) Profile from GEDI Level2B data.

usage
getLevel2BPAIProfile(level2b)

arguments
level2b A GEDI Level2B object (output of readLevel2B function). An S4 object of class "gedi.level2b".

value
Returns an S4 object of class data.table-class containing the elevation and relative heights.

see also
https://lpdaac.usgs.gov/products/gedi02_bv001/

examples
# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
    "GEDI02_B_201910080338_001964_T05337_02_001_01_sub.zip",
    package="rGEDI")

# Unzipping GEDI level2A data
level2Apath <- unzip(level2A_fp_zip,exdir = outdir)

# Reading GEDI level2A data (h5 file)
level2a<-readLevel2A(level2Apath=level2Apath)

# Extracting GEDI Elevation and Height Metrics
level2AM<-getLevel2AM(level2a)
head(level2AM)

close(level2a)
getLevel2BPAVDProfile

# Extracting GEDI Plant Area Index (PAI) Profile (GEDI Level2B)
level2BPAIProfile<-getLevel2BPAIProfile(level2b)
head(level2BPAIProfile)

close(level2b)

getLevel2BPAVDProfile Get GEDI Plant Area Volume Density (PAVD) Index Profile (GEDI Level2B)

Description
This function extracts the Plant Area Volume Density (PAVD) Profile from GEDI Level2B data.

Usage
getLevel2BPAVDProfile(level2b)

Arguments
level2b A GEDI Level2B object (output of readLevel2B function). An S4 object of class "gedi.level2b".

Value
Returns an S4 object of class data.table-class containing the Plant Area Volume Density Index.

See Also
https://lpdaac.usgs.gov/products/gedi02_bv001/

Examples
# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019100800338_001964_T05337_02_001_01_sub.zip",
  package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)

# Reading GEDI level2B data (h5 file)
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Extracting GEDI Plant Area Volume Density (PAVD) Index
level2BPAVDProfile<-getLevel2BPAVDProfile(level2b)
head(level2BPAVDProfile)

close(level2b)
getLevel2BVPM  

Get GEDI Canopy Cover and Vertical Profile Metrics (GEDI Level2B)

Description

This function extracts information from GEDI Level2B data: Total Plant Area Index, Foliage Height Diversity, Foliage Clumping Index, Total Gap Probability (theta), and Total canopy cover.

Usage

getLevel2BVPM(level2b)

Arguments

level2b  
A GEDI Level2B object (output of readLevel2B function). An S4 object of class "gedi.level2b".

Details

These are the biophysical variables extracted:

- pai  Total Plant Area Index.
- fhd_normal Foliage Height Diversity.
- omega Foliage Clumping Index.
- pgap_theta Total Gap Probability (theta).
- cover Total canopy cover.

Value

Returns an S4 object of class data.table-class containing the Vegetation Profile Biophysical Variables.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv001/

Examples

# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
"GEDI02_B_2019108080338_001964_T05337_02_001_01_sub.zip",
package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)

# Reading GEDI level2B data (h5 file)
```r
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Extracting GEDI Vegetation Profile Biophysical Variables
level2BVPM<-getLevel2BVPM(level2b)
head(level2BVPM)

close(level2b)
```

---

**gridStatsLevel2AM**

*Compute Grids with Descriptive Statistics of GEDI derived Elevation and Height Metrics (Level2A)*

**Description**

This function computes a series of user defined descriptive statistics within each grid cell for GEDI derived Elevation and Height Metrics (Level2A).

**Usage**

```r
gridStatsLevel2AM(level2AM, func, res)
```

**Arguments**

- `level2AM`: A GEDI Level2AM object (output of `getLevel2AM` function). An S4 object of class "data.table".
- `func`: The function(s) to be applied to each cell
- `res`: Spatial resolution in decimal degrees for the output raster layer

**Value**

Return a raster layer(s) of selected GEDI Elevation and Height Metric(s)

**See Also**

https://lpdaac.usgs.gov/products/gedi02_av001/

**Examples**

```r
# specify the path to GEDI level2A data (zip file)
outdir = tempdir()
level2A_fp_zip <- system.file("extdata",
    "GEDI02_A_2019108080338_001964_T05337_02_001_01_sub.zip",
    package="rGEDI")

# Unzipping GEDI level2A data
level2Apath <- unzip(level2A_fp_zip,exdir = outdir)

# Reading GEDI level2A data (h5 file)
level2a<-readLevel2A(level2Apath=level2Apath)
```
# Get GEDI derived Elevation and Height Metrics
level2AM <- getLevel2AM(level2a)
head(level2AM)

#' Define your own function
def mySetOfMetrics = function(x)
{
  metrics = list(
    min = min(x), # Min of z
    max = max(x), # Max of z
    mean = mean(x), # Mean of z
    sd = sd(x)# Sd of z
  )
  return(metrics)
}

#' Computing a serie of GEDI metrics
ZTstats <- gridStatsLevel2AM(level2AM = level2AM, func = mySetOfMetrics(elev_highestreturn), res = 0.005)
plot(ZTstats)

#' Computing the maximum of RH100 only
maxRH100 <- gridStatsLevel2AM(level2AM = level2AM, func = max(rh100), res = 0.005)
plot(maxRH100)

#' Computing the mean of ZG only
ZGmean <- gridStatsLevel2AM(level2AM = level2AM, func = mean(elev_lowestmode), res = 0.005)
plot(ZGmean)
close(level2a)

---

**gridStatsLevel2BVPM**  
*Compute Grids with Descriptive Statistics of GEDI derived Canopy Cover and Vertical Profile Metrics (Level2B)*

**Description**

This function computes a series of user defined descriptive statistics within each grid cell for GEDI derived Canopy Cover and Vertical Profile Metrics (Level2B)

**Usage**

gridStatsLevel2BVPM(level2BVPM, func, res)

**Arguments**

- **level2BVPM**: A GEDI Level2AM object (output of `getLevel2BVPM` function). An S4 object of class "data.table".
- **func**: The function(s) to be applied to each cell
- **res**: Spatial resolution in decimal degrees for the output raster layer
gridStatsLevel2BVPM

Value

Returns a raster layer(s) of selected GEDI Canopy Cover and Vertical Profile Metric(s)

See Also

https://lpdaac.usgs.gov/products/gedi02_bv001/

Examples

```r
# specify the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019100800338_001964_T05337_02_001_01_sub.zip",
  package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)

# Reading GEDI level2B data (h5 file)
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Get GEDI derived Canopy Cover and Vertical Profile Metrics
level2BVPM<-getLevel2BVPM(level2b)
head(level2BVPM)

# Define your own function
mySetOfMetrics = function(x)
{
  metrics = list(
    min =min(x), # Min of z
    max = max(x), # Max of z
    mean = mean(x), # Mean of z
    sd = sd(x)# Sd of z
  )
  return(metrics)
}

# Computing a serie of statistics of GEDI derived canopy cover
cover_stats<-gridStatsLevel2BVPM(level2BVPM = level2BVPM, func=mySetOfMetrics(cover), res=0.005)
plot(cover_stats)

# Computing the max of the Total Plant Area Index only
pai_max<-gridStatsLevel2BVPM(level2BVPM = level2BVPM, func=max(pai), res=0.005)
plot(pai_max)

# Computing the Foliage Height Diversity Index only
fhd_mean<-gridStatsLevel2BVPM(level2BVPM = level2BVPM, func=mean(fhd_normal), res=0.005)
plot(fhd_mean)

close(level2b)
```
Description

For gedi.fullwaveform: will plot the full waveform

for gedi.level1bSim: will plot the simulated waveform

Usage

plot(x, y, ...)

## S4 method for signature 'gedi.fullwaveform,missing'
plot(x, relative = FALSE, polygon = FALSE, ...)

## S4 method for signature 'gedi.level1bSim,missing'
plot(x, relative = FALSE, polygon = FALSE, method = "RXWAVEINT", ...)

Arguments

x An object of class "gedi.fullwaveform". (output of getLevel1BWF function)
y not used (inherited from R base)
... will be passed to the main plot
relative if TRUE, the Waveform Amplitude will be showed in percentage (%)
polygon if TRUE, the polygon will be added to the plot
method methods used for simulating the GEDI full-waveform ("RXWAVEINT","RXWAVEINT" or "RXWAVECOUNT"). Default is "RXWAVECOUNT".

Value

No return value

Examples

# Specifying the path to GEDI level1B data (zip file)
outdir = tempdir()
level1B_fp_zip <- system.file("extdata",
   "GEDI01_B_2019108080338_001964_T05337_02_003_01_sub.zip",
   package="rGEDI")

# Unzipping GEDI level1B data
level1Bpath <- unzip(level1B_fp_zip,exdir = outdir)

# Reading GEDI level1B data (h5 file)
plot

levellb<-readLevel1B(level1Bpath=level1Bpath)

# Extracting GEDI Full-Waveform
wf <- getLevel1BWF(level1b, shot_number="19640521100108408")

# Plotting GEDI Full-waveform
oldpar<-par()
par(mfrow = c(1,2), cex.axis = 1.5)
plot(wf, relative=FALSE, polygon=TRUE, type="l", lwd=2, col="forestgreen", xlab="", ylab="Elevation (m)")
plot(wf, relative=TRUE, polygon=TRUE, type="l", lwd=2, col="forestgreen", xlab="Waveform Amplitude (%)", ylab="Elevation (m)")
par(oldpar)
close(level1b)

outdir <- tempdir()
zipfile_amazon <- system.file("extdata", "Amazon.zip", package="rGEDI")
zipfile_Savanna <- system.file("extdata", "Savanna.zip", package="rGEDI")

lasfile_amazon <- unzip(zipfile_amazon,exdir=outdir)
lasfile_Savanna <- unzip(zipfile_Savanna,exdir=outdir)

# Reading and plot ALS file
libsAvailable = require(lidR) && require(plot3D)
if (libsAvailable) {
  las_amazon<-readLAS(lasfile_amazon)
las_Savanna<-readLAS(lasfile_Savanna)
}

# Extracting plot center geolocations
xcenter_amazon = mean(las_amazon@bbox[1,])
ycenter_amazon = mean(las_amazon@bbox[2,])
xcenter_Savanna = mean(las_Savanna@bbox[1,])
ycenter_Savanna = mean(las_Savanna@bbox[2,])

# Simulating GEDI full-waveform
wf_amazon<-gediWFSimulator(
  input=lasfile_amazon,
  output=file.path(
    outdir,
    "gediWF_amazon_simulation.h5"
  ),
  coords = c(xcenter_amazon, ycenter_amazon))
wf_Savanna<-gediWFSimulator(
  input=lasfile_Savanna,
  output=file.path(
    outdir,
    "gediWF_Savanna_simulation.h5"
  ),
  coords = c(xcenter_Savanna, ycenter_Savanna))

# Plot Full-waveform
par(mfrow=c(2,2), mar=c(4,4,0,0), oma=c(0,0,1,1),cex.axis = 1.2)
plotPAIProfile

Plot GEDI Plant Area Index (PAI) Profile

Description
This functions plots Plant Area Index (PAI) Profile (GEDI level2B)

Usage
plotPAIProfile(level2BPAIProfile, beam = "BEAM0101", elev = TRUE)

Arguments
level2BPAIProfile
A GEDI Level2B object (output of getLevel2BPAIProfile function). An S4 object of class “data.table”.

beam
Select GEDI beam. Default is "BEAM0101”. See details section.

elev
If TRUE, elevation will be used for plotting the PAI profile. Otherwise, height will be used instead.
plotPAIProfile

Details

list of GEDI beams. See the output of `getLevel2BPAIProfile` function.

- `BEAM0000`
- `BEAM0001`
- `BEAM0010`
- `BEAM0011`
- `BEAM0101`
- `BEAM0110`
- `BEAM1000`
- `BEAM1011`

Value

Returns a ggplot object. See `ggplot` package.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv001/

Examples

# specify the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019108080338_001964_T05337_02_001_01_sub.zip",
  package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)

# Reading GEDI level2B data (h5 file)
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Get Plant Area Volume Density profile
level2BPAIProfile<-getLevel2BPAIProfile(level2b)

# Plot Level2B PAI Profile
gprofile<-plotPAIProfile(level2BPAIProfile, beam="BEAM0101", elev=TRUE)

close(level2b)
Description

This functions plots Plant Area Volume Density profile (GEDI level2B)

Usage

plotPAVDProfile(level2BPAVDProfile, beam = "BEAM0101", elev = TRUE)

Arguments

level2BPAVDProfile
A GEDI Level2B object (output of getLevel2BPAVDProfile function). An S4 object of class "data.table".

beam
Select GEDI beam. Default is "BEAM0101". See details section.

elev
If TRUE, elevation will be used for plotting the PAVD profile. Otherwise, height will be used instead.

Details

list of GEDI beams. See the output of getLevel2BPAVDProfile function.

- BEAM0000
- BEAM0001
- BEAM0010
- BEAM0011
- BEAM0101
- BEAM0110
- BEAM1000
- BEAM1011

Value

Returns a ggplot object. See ggplot package.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv001/
plotWFMetrics

Example

# specify the path to GEDI level2B data (zip file)
outdir <- tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_201910080338_001964_T05337_02_001_01_sub.zip",
  package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)

# Reading GEDI level2B data (h5 file)
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Get Plant Area Volume Density profile
level2BPAVDProfile<-getLevel2BPAVDProfile(level2b)

# Plot Level2B PAVD Profile
gprofile<-plotPAVDProfile(level2BPAVDProfile, beam="BEAM0101", elev=TRUE)

close(level2b)

plotWFMetrics GEDI full waveform plot with metrics

Description

Plots the waveform with overlaid RH metrics

Usage

plotWFMetrics(level1b, level2a, shot_number, rh=c(25, 50, 75),...)

Arguments

- **level1b**: A GEDI Level1B object (output of `readLevel1B` function). An S4 object of class "gedi.level1b".
- **level2a**: A GEDI Level2A object (output of `readLevel2A` function). An S4 object of class "gedi.level2a".
- **shot_number**: Shot number. A scalar representing the shot number of a giving pulse.
- **rh**: Integer vector. Specify which RH metrics to plot except rh0 and rh100, default c(25, 50, 75).
  
Value

Returns a raster layer(s) of selected GEDI Canopy Cover and Vertical Profile Metric(s)
polyStatsLevel2AM

# polyStatsLevel2AM

## Description

Computes a Series of Statistics from GEDI derived Elevation and Height Metrics (Level2A) within a given area defined or not by a polygon.

## Usage

```r
polyStatsLevel2AM(level2AM, func, id=NULL)
```

## Arguments

- `level2AM`: A GEDI Level2AM object (output of `getLevel2AM` function). An S4 object of class "data.table".
- `func`: The function to be applied for computing the defined statistics.
- `id`: A vector containing the polygon id for each GEDI observation. Default is NULL.

## Examples

```r
# specify the path to GEDI level1B and Level2A data (zip file)
outdir = tempdir()
level1B_fp_zip <- system.file("extdata",
    "GEDI01_B_2019108080338_001964_T05337_02_003_01_sub.zip",
    package="rGEDI")
level2A_fp_zip <- system.file("extdata",
    "GEDI02_A_2019108080338_001964_T05337_02_001_01_sub.zip",
    package="rGEDI")

# Unzipping GEDI level1B data
level1Bpath <- unzip(level1B_fp_zip,exdir = outdir)
level2Apath <- unzip(level2A_fp_zip,exdir = outdir)

# Reading GEDI level1B and Level2A data (h5 file)
level1b<-readLevel1B(level1Bpath=level1Bpath)
level2a<-readLevel2A(level2Apath=level2Apath)

shot_number = "19640521100108408"
plotWFMetrics(level1b, level2a, shot_number, rh=c(25, 50, 75, 90))
close(level1b)
close(level2a)
```
Value

Returns an S4 object of class \texttt{data.table-class} containing Statistics of GEDI level2A defined metrics.

See Also

https://lpdaac.usgs.gov/products/gedi02_av001/

Examples

```r
# Specifying the path to GEDI level2A data (zip file)
outdir = tempdir()
level2A_fp_zip <- system.file("extdata", "GEDI02_A_2019108080338_001964_T05337_02_001_01_sub.zip", package="rGEDI")

# Unzipping GEDI level2A data
level2Apath <- unzip(level2A_fp_zip,exdir = outdir)

# Reading GEDI level2A data (h5 file)
level2a<-readLevel2A(level2Apath=level2Apath)

# Specifying the path to shapefile
polygon_filepath <- system.file("extdata", "stands_cerrado.shp", package="rGEDI")

# Reading shapefile as SpatialPolygonsDataFrame object
library(raster)
polygon_spdf<-shapefile(polygon_filepath)

# Extracting GEDI Elevation and Relative Metrics (level2A)
level2AM<-getLevel2AM(level2a)
head(level2AM)

# Clipping GEDI data by geometry
level2AM_clip = clipLevel2AMGeometry(level2AM, polygon_spdf, split_by="id")

# Define your own function
mySetOfMetrics = function(x)
{
  metrics = list(
    min = min(x), # Min of x
    max = max(x), # Max of x
    mean = mean(x), # Mean of x
    sd = sd(x)# Sd of x
  )
  return(metrics)
}

# Computing the maximum of RH100
RH100max<-polyStatsLevel2AM(level2AM_clip,func=max(rh100), id=NULL)

# Computing the maximum of RH100 stratified by polygon
```
RH100max_poly<-polyStatsLevel2AM(level2AM_clip,func=max(rh100), id=NULL)

# Computing a serie statistics for GEDI metrics stratified by polygon
RH100metrics<-polyStatsLevel2AM(level2AM_clip,func=mySet0FMetrics(rh100), id=level2AM_clip$id)

close(level2a)

polyStatsLevel2BVPM

Compute descriptive statistics of GEDI Canopy Cover and Vertical Profile Metrics

Description
Computes a Series of Statistics of GEDI derived Canopy Cover and Vertical Profile metrics within a given area defined or not by a polygon

Usage
polyStatsLevel2BVPM(level2BVPM, func, id=NULL)

Arguments
level2BVPM A GEDI Level2BVPM object (output of getLevel2BVPM function). An S4 object of class "data.table".
func The function to be applied for computing the defined statistics
id A vector containing the polygon id for each GEDI observation. Default is NULL

Value
Returns an S4 object of class data.table-class containing Statistics of GEDI level2BVPM defined metrics

See Also
https://lpdaac.usgs.gov/products/gedi02_bv001/

Examples
# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
"GEDI02_B_201910080338_001964_T05337_02_001_01_sub.zip",
package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip,exdir = outdir)
# Reading GEDI level2B data (h5 file)
level2b<-readLevel2B(level2Bpath=level2Bpath)

# Specifying the path to shapefile
polygon_filepath <- system.file("extdata", "stands_cerrado.shp", package="rGEDI")

# Reading shapefile as SpatialPolygonsDataFrame object
library(raster)
polygon_spdf<-shapefile(polygon_filepath)

# Extracting GEDI Canopy Cover and Vertical Profile Metrics
level2BVPM<-getLevel2BVPM(level2b)
head(level2BVPM)

# Clipping GEDI data by geometry
level2BVPM_clip = clipLevel2BVPMGeometry(level2BVPM, polygon_spdf, split_by="id")

# Define your own function
mySetOfMetrics = function(x)
{
metrics = list(
    min =min(x), # Min of x
    max = max(x), # Max of x
    mean = mean(x), # Mean of x
    sd = sd(x)# Sd of x
)
return(metrics)
}

# Computing the max of the Total Plant Area Index
pai_max<-polyStatsLevel2BVPM(level2BVPM_clip,func=max(pai), id=NULL)
pai_max

# Computing the max of the Total Plant Area Index stratified by polygon
pai_max_poly<-polyStatsLevel2BVPM(level2BVPM_clip,func=max(pai), id="poly_id")
head(pai_max_poly)

# Computing the serie of statistics of canopy cover stratified by polygon
cover_metrics<-polyStatsLevel2BVPM(level2BVPM_clip,func=mySetOfMetrics(cover),
    id=level2BVPM_clip$id)

head(cover_metrics)
close(level2b)

---

**readLevel1B**

*Read GEDI Level1B data (Geolocated Waveforms)*

**Description**

This function reads GEDI level1B products: geolocated Waveforms
Usage

readLevel1B(level1Bpath)

Arguments

level1Bpath  File path pointing to GEDI level1B data. Data in HDF5 Hierarchical Data Format (.h5).

Value

Returns an S4 object of class "gedi.level1b" containing GEDI level1B data.

See Also

hdf5r::H5File in the hdf5r package and https://lpdaac.usgs.gov/products/gedi01_bv001/

Examples

# Specifying the path to GEDI level1B data (zip file)
outdir = tempdir()
level1B_fp_zip <- system.file("extdata",
  "GEDI01_B_2019108080338_001964_T05337_02_003_01_sub.zip",
  package="rGEDI")
# Unzipping GEDI level1B data
level1Bpath <- unz(level1B_fp_zip,exdir = outdir)
# Reading GEDI level1B data (h5 file)
level1b<-readLevel1B(level1Bpath=level1Bpath)
close(level1b)

readLevel2A

Read GEDI Level2A data (Basic Full Waveform derived Metrics)

Description

This function reads GEDI level2A products: ground elevation, canopy top height, and relative heights (RH).

Usage

readLevel2A(level2Apath)

Arguments

level2Apath  File path pointing to GEDI level2A data. Data in HDF5 Hierarchical Data Format (.h5).
Value

Returns an S4 object of class ’gedi.level2a’ containing GEDI level2A data.

See Also

https://lpdaac.usgs.gov/products/gedi02_av001/

Examples

```r
# Specifying the path to GEDI level2A data (zip file)
outdir = tempdir()
level2A_fp_zip <- system.file("extdata",
  "GEDI02_A_20191080338_001964_T05337_02_001_01_sub.zip",
  package="rGEDI")

# Unzipping GEDI level2A data
level2Apath <- unzip(level2A_fp_zip,exdir = outdir)

# Reading GEDI level2A data (h5 file)
level2a<-readLevel2A(level2Apath=level2Apath)

close(level2a)
```

---

**readLevel2B**

**Read GEDI Level2B data (Biophysical Variables)**

**Description**

This function reads GEDI level2B products: canopy cover, Plant Area Index (PAI), Plant Area Volume Density (PAVD), and Foliage Height Diversity (FHD).

**Usage**

```r
readLevel2B(level2Bpath)
```

**Arguments**

- `level2Bpath`  
  File path pointing to GEDI level2B data. Data in HDF5 Hierarchical Data Format (.h5).

**Value**

Returns an S4 object of class ’gedi.level2b’ containing GEDI level2B data.
Examples

# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
    "GEDI02_B_2019108080338_001964_T05337_02_001_01_sub.zip",
    package="rGEDI")

# Unzipping GEDI level2A data
level2Bpath <- unzip(level2B_fp_zip, exdir = outdir)

# Reading GEDI level2B data (h5 file)
level2b <- readLevel2B(level2Bpath = level2Bpath)

close(level2b)
Index

clipLevel1B, 4
clipLevel1BGeo, 5
clipLevel1BGeoGeometry, 6
clipLevel1BGeometry, 8
clipLevel2A, 9
clipLevel2AGeometry, 10
clipLevel2AM, 12
clipLevel2AMGeometry, 13
clipLevel2B, 14
clipLevel2BGeometry, 16
clipLevel2BPAIProfile, 17
clipLevel2BPAIProfileGeometry, 18
clipLevel2BPAVDProfile, 20
clipLevel2BPAVDProfileGeometry, 21
clipLevel2BVPM, 22
clipLevel2BVPMGeometry, 24
close, 25
close, gedi.level1b-method (close), 25
close, gedi.level1bSim-method (close), 25
close, gedi.level2a-method (close), 25
close, gedi.level2b-method (close), 25
gedi.fullwaveform
  (gedi.fullwaveform-class), 26
gedi.fullwaveform-class, 26
gedi.level1b (gedi.level1b-class), 26
gedi.level1b-class, 26
gedi.level1bSim, 32
gedi.level1bSim
  (gedi.level1bSim-class), 27
gedi.level1bSim-class, 27
gedi.level2a (gedi.level2a-class), 27
gedi.level2a-class, 27
gedi.level2b (gedi.level2b-class), 28
gedi.level2b-class, 28
gediDownload, 28
gedefinder, 29
gediWFMetrics, 31
gediWFSimulator, 38
getLevel1BGeo, 41
getLevel1BWF, 45, 53
getAddress1B, 4,
getAddress1BGeo, 5
getLevel1BGeoGeometry, 6
getLevel1BGeometry, 8
getAddress2A, 9
getLevel2AGeometry, 10
getLevel2AM, 12
getLevel2AMGeometry, 13
getLevel2B, 14
getLevel2BGeometry, 16
getLevel2BPAIProfile, 17
getLevel2BPAIProfileGeometry, 18
getLevel2BPAVDProfile, 20
getLevel2BPAVDProfileGeometry, 21
getLevel2BVPM, 22
getLevel2BVPMGeometry, 24
hdf5r::H5File, 26–28, 40, 63
plot, 53
plot, gedi.fullwaveform, missing-method (plot), 53
plot, gedi.level1bSim, missing-method (plot), 53
plotPAIProfile, 55
plotPAVDProfile, 57
plotWFMetrics, 58
polyStatsLevel2AM, 59
polyStatsLevel2BVPM, 61
raster::shapefile(), 7, 8, 11, 13, 16, 19, 21, 24
readLevel1B, 4, 5, 7, 8, 22, 24, 41, 45, 58, 62
readLevel2A, 9, 11–13, 46, 58, 63
readLevel2B, 15, 16, 47–49, 64
rGEDI-package, 3

66