Package: rGEDI (via r-universe)

October 15, 2024

Type Package

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

Version 0.5.0

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, RColorBrewer, sp, stats, stars

Suggests lattice, leaflet, leafsync, lidR, plot3D, rasterVis, sf, viridis

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

BugReports https://github.com/carlos-alberto-silva/rGEDI/issues

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

Repository https://carlos-alberto-silva.r-universe.dev

RemoteUrl https://github.com/carlos-alberto-silva/rGEDI

RemoteRef HEAD

RemoteSha 822f4a360823cf8cff6b96808875072508ece122

Contents

`rGEDI-package` *rGEDI: An R Package for NASA's Global Ecosystem Dynamics Investigation (GEDI) Data Visualizing and Processing.*

Description

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 https://github.com/carlos-alberto-silva/rGEDI/blob/master/README.md

---

clipLevel1B *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.
`ymax`	Numeric. North latitude (y) coordinate of the bounding rectangle, in decimal degrees.
`output`	Optional character path where to save the new hdf5file. 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_bv002/

Examples

```
# Specifying the path to GEDI level1B data (zip file)
outdir <- tempdir()

level1B_fp_zip <- system.file(”extdata”,
  ”GEDI01_B_2019108080338_O01964_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_2019108080338_O01964_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 (`getLevel1BGeo()`) data a within given bounding coordinates

Usage

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

Arguments

`level1BGeo` A `data.table::data.table` resulting from `getLevel1BGeo()`.

`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::data.table`.

See Also

https://lpdaac.usgs.gov/products/gedi01_bv002/

Examples

```
# Specifying the path to GEDI level1B data (zip file)
outdir <- tempdir()
level1B_fp_zip <- system.file("extdata",
  "GEDI01_B_2019108080338_O01964_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)

hasLeaflet <- require(leaflet)

if (hasLeaflet) {
  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 level1BGeo extracted geolocation (level1BGeo) data within a given geometry

Usage

```
clipLevel1BGeoGeometry(level1BGeo, polygon, split_by = "id")
```

Arguments

`level1BGeo`	A `data.table::data.table` resulting from `getLevel1BGeo()` function.
`polygon`	Polygon. An object of class `sf::sf`, which can be loaded as an ESRI shapefile using sf::st_read function in the *sf* 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::data.table` containing the clipped GEDI level1B extracted geolocations.

See Also

https://lpdaac.usgs.gov/products/gedi01_bv002/

Examples

```
# Specifying the path to GEDI level1B data (zip file)
outdir <- tempdir()
level1B_fp_zip <- system.file("extdata",
  "GEDI01_B_2019108080338_O01964_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 sf object
library(sf)
polygon <- sf::st_read(polygon_filepath)

# Clipping GEDI Full Waveform Geolocations by Geometry
level1BGeo_clip <- clipLevel1BGeoGeometry(level1BGeo, polygon, split_by = ”id”)

hasLeaflet <- require(leaflet)

if (hasLeaflet) {
  leaflet() %>%
    addCircleMarkers(level1BGeo_clip$longitude_bin0,
      level1BGeo_clip$latitude_bin0,
      radius = 1,
      opacity = 1,
      color = ”red”
    ) %>%
    addScaleBar(options = list(imperial = FALSE)) %>%
    addPolygons(
      data = polygon, weight = 1, col = ”white”,
      opacity = 1, fillOpacity = 0
    ) %>%
    addProviderTiles(providers$Esri.WorldImagery)
}

close(level1b)
```

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, output = "", split_by = NULL)
```

Arguments

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

`polygon` Polygon or Multipolygon. An object opened with `sf::st_read`,

`output` Optional character path where to save the new `hdf5r::H5File`. 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_O01964_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)

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

# Reading shapefile as sf object
library(sf)
polygon <- sf::st_read(polygon_filepath)

# Sepecifing output file and path
output <- file.path(outdir, "GEDI01_B_2019108080338_O01964_T05337_02_003_01_clip")

# clipping GEDI Level1B data by extent boundary box
level1b_clip <- clipLevel1BGeometry(level1b,
  polygon = polygon,
  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.

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

`output` Optional character path where to save the new hdf5file. 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_av002/

Examples

```
outdir <- tempdir()

# Specifying the path to GEDI level2A data (zip file)
level2A_fp_zip <- system.file("extdata",
  "GEDI02_A_2019108080338_O01964_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

print(level2a)

# Specifying output file and path
output <- file.path(outdir, "GEDI02_A_2019108080338_O01964_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)
```

`clipLevel2AGeometry` *Clip GEDI Level2A data by geometry*

Description

This function clips GEDI Level2A data within a given geometry

Usage

```
clipLevel2AGeometry(level2a, polygon, output = "", split_by = NULL)
```

Arguments

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

`polygon` Polygon. An object of class `sf::sf`, which can be loaded as an ESRI shapefile using `sf::st_read()` function in the *sf* 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_av002/

Examples

```
outdir <- tempdir()

# Specifying the path to GEDI level2A data (zip file)
level2A_fp_zip <- system.file("extdata",
  "GEDI02_A_2019108080338_O01964_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 sf object
library(sf)
polygon <- sf::st_read(polygon_filepath)

# Specifying output file and path
output <- file.path(outdir, "GEDI02_A_2019108080338_O01964_T05337_02_001_01_clip")

# Clipping GEDI Level2A data by geometry
level2a_clip <- clipLevel2AGeometry(level2a,
  polygon = polygon,
  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::data.table containing the clipped elevation and relative heights metrics.

See Also

`https://lpdaac.usgs.gov/products/gedi02_av002/`

Examples

```
# Specifying the path to GEDI level2A data (zip file)
outdir <- tempdir()
level2A_fp_zip <- system.file("extdata",
  "GEDI02_A_2019108080338_O01964_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

```
clipLevel2AMGeometry(level2AM, polygon, split_by = "id")
```

Arguments

`level2AM`	A GEDI Level2A object (output of `readLevel2A()` function). An S4 object of class "data.table".
`polygon`	Polygon. An object of class `sf::sf`, which can be loaded as an ESRI shapefile using sf::st_read function in the *sf* 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::data.table containing the clipped elevation and relative heights metrics.

Examples

```
# Specifying the path to GEDI level2A data (zip file)
outdir <- tempdir()
level2A_fp_zip <- system.file("extdata",
  "GEDI02_A_2019108080338_O01964_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 sf object
```

```
library(sf)
polygon <- sf::st_read(polygon_filepath)

# Clipping GEDI data by Geometry
level2AM_clip <- clipLevel2AMGeometry(level2AM, polygon, split_by = "id")

hasLeaflet <- require(leaflet)

if (hasLeaflet) {
  leaflet() %>%
    addCircleMarkers(level2AM_clip$lat_lowestmode,
      level2AM_clip$lon_lowestmode,
      radius = 1,
      opacity = 1,
      color = "red"
    ) %>%
    addScaleBar(options = list(imperial = FALSE)) %>%
    addPolygons(
      data = polygon, 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_bv002/

Examples

```
outdir <- tempdir()

# Specifying the path to GEDI level2B data (zip file)
level2B_fp_zip <- system.file(”extdata”,
  ”GEDI02_B_2019108080338_O01964_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
output <- file.path(outdir, ”GEDI02_B_2019108080338_O01964_T05337_02_001_01_clip”)

# Clipping GEDI data by extent boundary box
level2b_clip <- clipLevel2B(level2b, xmin, xmax, ymin, ymax)

close(level2b)
close(level2b_clip)
```

`clipLevel2BGeometry` *Clip GEDI Level2B data by geometry*

Description

This function extracts GEDI Level1B data within a given geometry

Usage

```
clipLevel2BGeometry(level2b, polygon, output = "", split_by = NULL)
```

Arguments

`level2b`	A GEDI Level2B object (output of `readLevel2B()` function). An S4 object of class "gedi.level2b".
`polygon`	Polygon. An object of class `sf::sf`, which can be loaded as an ESRI shapefile using sf::st_read function in the *sf* 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_bv002/

Examples

```
outdir <- tempdir()

# Specifying the path to GEDI level2B data (zip file)
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019108080338_O01964_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 sf object
library(sf)
polygon <- sf::st_read(polygon_filepath)

# Specifying output file and path
output <- file.path(outdir, "GEDI02_B_2019108080338_O01964_T05337_02_001_01_clip")
```

```
# Clipping GEDI data by extent boundary box
level2b_clip <- clipLevel2BGeometry(level2b,
  polygon = polygon,
  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

```
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.

Value

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

See Also

`https://lpdaac.usgs.gov/products/gedi02_bv002/`

Examples

```
# Specifying the path to GEDI level2B data (zip file)
outdir <- tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019108080338_O01964_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, split_by = NULL)
```

Arguments

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

`polygon`
: Polygon. An object of class `sf::sf`, which can be loaded as an ESRI shapefile using sf::st_read function in the *sf* 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::data.table containing the Plant Area Index profile data.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv002/

Examples

```
# Specifying the path to GEDI level2B data (zip file)
outdir <- tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019108080338_O01964_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)

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

# Reading shapefile as sf object
library(sf)
polygon <- sf::st_read(polygon_filepath)

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

close(level2b)
```

clipLevel2BPAVDProfile *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

```
clipLevel2BPAVDProfile(level2BPAVDProfile, xmin, xmax, ymin, ymax)
```

Arguments

`level2BPAVDProfile`
: A GEDI Level2B object (output of `getLevel2BPAVDProfile()` 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::data.table containing the Plant Area Volume Density profile data.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv002/

Examples

```
# specify the path to GEDI level2B data (zip file)
outdir <- tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019108080338_O01964_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 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, split_by = NULL)
```

Arguments

`level2BPAVDProfile`
: A GEDI Level2B object (output of `getLevel2BPAIProfile()` function). An S4 object of class "gedi.level2b".

`polygon`
: Polygon. An object of class `sf::sf`, which can be loaded as an ESRI shapefile using sf::st_read function in the *sf* 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::data.table containing the Plant Area Volume Density profile data.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv002/

Examples

```
# Specifying the path to GEDI level2B data (zip file)
outdir <- tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019108080338_O01964_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 profile
level2BPAVDProfile <- getLevel2BPAVDProfile(level2b)

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

# Reading shapefile as sf object
library(sf)
polygon <- sf::st_read(polygon_filepath)

# Clipping GEDI Plant Area Volume Density profile by geometry
level2BPAVDProfile_clip <- clipLevel2BPAVDProfileGeometry(
  level2BPAVDProfile,
  polygon,
  split_by = "id"
)

close(level2b)
```

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.
`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::data.table containing the Canopy Cover and Vertical Profile metrics.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv002/

Examples

```
# Specifying the path to GEDI level2B data (zip file)
outdir <- tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019108080338_O01964_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)

hasLeaflet <- require(leaflet)

if (hasLeaflet) {
  leaflet() %>%
    addCircleMarkers(level2b_clip$longitude_bin0,
      level2b_clip$latitude_bin0,
      radius = 1,
      opacity = 1,
      color = "red"
    ) %>%
    addScaleBar(options = list(imperial = FALSE)) %>%
    addProviderTiles(providers$Esri.WorldImagery)
}

close(level2b)
```

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

```
clipLevel2BVPMGeometry(level2BVPM, polygon, split_by = NULL)
```

Arguments

`level2BVPM`	A GEDI Level2B object (output of `readLevel1B()` function). An S4 object of class "gedi.level2b".
`polygon`	Polygon. An object of class `sf::sf`, which can be loaded as an ESRI shapefile using sf::st_read function in the *sf* 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::data.table containing the Canopy Cover and Vertical Profile metrics.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv002/

Examples

```
# Specifying the path to GEDI level2B data (zip file)
outdir <- tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019108080338_O01964_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)
```

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

# Reading shapefile as sf object
library(sf)
polygon <- sf::st_read(polygon_filepath)

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

hasLeaflet <- require(leaflet)

if (hasLeaflet) {
  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, weight = 1, col = "white",
      opacity = 1, fillOpacity = 0
    ) %>%
    addProviderTiles(providers$Esri.WorldImagery)
}

close(level2b)
```

close,gedi.level1b-method

Safely closes the gedi.level1b

Description

Closing files will avoid locking HDF5 GEDI files.

Closing files will avoid locking HDF5 GEDI files.

Closing files will avoid locking HDF5 GEDI files.

Usage

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

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

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

Arguments

`con`	An object of class `gedi.level2b`
`...`	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

`H5File` in the `hdf5r` package and `https://lpdaac.usgs.gov/products/gedi01_bv002/`

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

`H5File` in the `hdf5r` package and `https://lpdaac.usgs.gov/products/gedi02_av002/`

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

`H5File` in the `hdf5r` package and `https://lpdaac.usgs.gov/products/gedi02_bv002/`

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

```
gediDownload(
  filepath,
  outdir = NULL,
  overwrite = FALSE,
  buffer_size = 512,
  timeout = 10
)
```

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.

`timeout` integer; connection timeout in seconds.

Value

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

References

Credits to Cole Krehbiel. Code adapted from `https://git.earthdata.nasa.gov/projects/LPDUR/repos/daac_data_download_r/browse/DAACDataDownload.R`

Examples

```
## Not run:
# Set path to GEDI data
# herein we will only download xml metedata
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()

# Create .netrc file
netrc = file.path(outdir, ".netrc")
netrc_conn <- file(netrc)

writeLines(c("machine urs.earthdata.nasa.gov",
            sprintf("login %s", Sys.getenv("NASA_USER")),
            sprintf("password %s", Sys.getenv("NASA_PASSWORD"))
), netrc_conn)

close(netrc_conn)

#' Downloading GEDI data
gediDownload(filepath,outdir)

## End(Not run)
```

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

```
gedifinder(
  product,
  ul_lat,
  ul_lon,
  lr_lat,
  lr_lon,
  version = "002",
  daterange = NULL
)
```

Arguments

`product` GEDI data level; Options: "GEDI01_B", "GEDI02_A", "GEDI02_B", "GEDI03", "GEDI04_A", "GEDI04_A", "GEDI04_B"

`ul_lat` Numeric. Upper left (ul) corner coordinates, in lat (decimal degrees) for the bounding box of the area of interest.

`ul_lon` Numeric. Upper left (ul) corner coordinates, in lon (decimal degrees) for the bounding box of the area of interest.

`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 "002".

`daterange` Vector. Date range. Specify your start and end dates using ISO 8601 [YYYY]-[MM]-[DD]T[hh]:[mm]:[ss]Z. Ex.: c("2019-07-01T00:00:00Z","2020-05-22T23:59:59Z"). 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 CMR tool: https://cmr.earthdata.nasa.gov/search/site/docs/search/api.html

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 = "002",
  daterange = daterange
)
```

getLevel1BGeo *Get GEDI Full Waveform Geolocations (GEDI Level1B)*

Description

This function extracts Pulse Full Waveform Geolocations from GEDI `gedi.level1b` data

Usage

```
getLevel1BGeo(level1b, select)
```

Arguments

`level1b`	A `gedi.level1b` object (output of `getLevel1BGeo()` function).
`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

Additional fields to be extracted from GEDI level 1B:

- *all_samples_sum* Sum of all values within the 10 km range window.
- *beam* Beam number 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_pulseflag* 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.
- *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 longerperiod 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::data.table containing the GEDI Full Waveform Geolocations

See Also

https://lpdaac.usgs.gov/products/gedi01_bv002/

Examples

```
# specify the path to GEDI level1B data (zip file)
outdir = tempdir()
level1B_fp_zip <- system.file("extdata",
                   "GEDI01_B_2019108080338_O01964_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 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_bv002/

Examples

```
# Specifying the path to GEDI level1B data (zip file)
outdir = tempdir()
level1B_fp_zip <- system.file("extdata",
                  "GEDI01_B_2019108080338_O01964_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".

Details

Characteristics. Flag indicating likely invalid waveform (1=valid, 0=invalid).

- *beam* Beam identifier
- *shot_number* Shot number
- *degrade_flag* Flag indicating degraded state of pointing and/or positioning information
- *quality_flag* Flag simplifying selection of most useful data
- *delta_time* Transmit time of the shot since Jan 1 00:00 2018
- *sensitivity* Maxmimum canopy cover that can be penetrated
- *solar_elevation* Solar elevation
- *lat_lowestmode* Latitude of center of lowest mode
- *lon_lowestmode* Longitude of center of lowest mode
- *elev_highestreturn* Elevation of highest detected return relative to reference ellipsoid Meters
- *elev_lowestmode* Elevation of center of lowest mode relative to reference ellipsoid
- *rh* Relative height metrics at 1% interval

Value

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

See Also

https://lpdaac.usgs.gov/products/gedi02_av002/

Examples

```
# Specifying the path to GEDI level2A data (zip file)
outdir = tempdir()
level2A_fp_zip <- system.file("extdata",
                  "GEDI02_A_2019108080338_O01964_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)
```

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".

Details

Characteristics. Flag indicating likely invalid waveform (1=valid, 0=invalid).

- *beam* Beam identifier
- *shot_number* Shot number
- *algorithmrun_flag* The L2B algorithm is run if this flag is set to 1 indicating data have sufficient waveform fidelity for L2B to run
- *l2b_quality_flag* L2B quality flag
- *delta_time* Transmit time of the shot since Jan 1 00:00 2018
- *lat_lowestmode* Latitude of center of lowest mode
- *lon_lowestmode* Longitude of center of lowest mode
- *elev_highestreturn* Elevation of highest detected return relative to reference ellipsoid
- *elev_lowestmode* Elevation of center of lowest mode relative to reference ellipsoid
- *height_lastbin* Height of the last bin of the pgap_theta_z, relative to the ground
- *pai_z* Plant Area Index profile

Value

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

See Also

`https://lpdaac.usgs.gov/products/gedi02_bv002/`

Examples

```
# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
                  "GEDI02_B_2019108080338_O01964_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 (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".

Details

Characteristics. Flag indicating likely invalid waveform (1=valid, 0=invalid).

- *beam* Beam identifier
- *shot_number* Shot number
- *algorithmrun_flag* The L2B algorithm is run if this flag is set to 1 indicating data have sufficient waveform fidelity for L2B to run
- *l2b_quality_flag* L2B quality flag
- *delta_time* Transmit time of the shot since Jan 1 00:00 2018
- *lat_lowestmode* Latitude of center of lowest mode

- *lon_lowestmode* Longitude of center of lowest mode
- *elev_highestreturn* Elevation of highest detected return relative to reference ellipsoid
- *elev_lowestmode* Elevation of center of lowest mode relative to reference ellipsoid
- *height_lastbin* Height of the last bin of the pgap_theta_z, relative to the ground
- *pavd_z* Plant Area Volume Density profile

Value

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

See Also

https://lpdaac.usgs.gov/products/gedi02_bv002/

Examples

```
# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
                  "GEDI02_B_2019108080338_O01964_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,
 cols = c("beam", "shot_number", "algorithmrun_flag", "l2b_quality_flag", "delta_time",
    "sensitivity", "solar_elevation", "latitude_lastbin", "latitude_bin0",
   "longitude_bin0", "longitude_lastbin", "elev_highestreturn", "elev_lowestmode",
    "rh100", "pai", "fhd_normal", "omega", "pgap_theta", "cover")
)
```

Arguments

`level2b`	A GEDI Level2B object (output of `readLevel2B()` function). An S4 object of class "gedi.level2b".
`cols`	A character vector containing the list of columns to be extracted. See the default columns in the description.

Details

These are the biophysical variables and additional information extracted by default:

- *beam* Beam identifier
- *shot_number* Shot number
- *algorithmrun_flag* The L2B algorithm is run if this flag is set to 1 indicating data have sufficient waveform fidelity for L2B to run
- *l2b_quality_flag* L2B quality flag
- *delta_time* Transmit time of the shot since Jan 1 00:00 2018
- *sensitivity* Maxmimum canopy cover that can be penetrated
- *solar_elevation* Solar elevation
- *latitude_lastbin* Latitude of last bin of the pgap_theta_z, interpolated from L1B waveform coordinate
- *latitude_bin0* Latitude of first bin of the pgap_theta_z, interpolated from L1B waveform coordinate
- *elev_highestreturn* Elevation of highest detected return relative to reference ellipsoid
- *elev_lowestmode* Elevation of center of lowest mode relative to reference ellipsoid
- *rh100* RH100 slice
- *pai* Total Plant Area Index
- *fhd_normal* Foliage Height Diversity
- *omega* Foliage Clumping Index
- *pgap_theta* Total Gap Probability (theta)
- *cover* Total canopy cover

Every other columns in the GEDI2B product are also available, you can specify each column by using the `cols` parameter.

Value

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

See Also

https://lpdaac.usgs.gov/products/gedi02_bv002/

Examples

```
# Specifying the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
                  "GEDI02_B_2019108080338_O01964_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 Vegetation Profile Biophysical Variables
level2BVPM<-getLevel2BVPM(level2b, cols=c("beam", "shot_number"))
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

```
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 stars raster layer

Value

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

See Also

https://lpdaac.usgs.gov/products/gedi02_av002/

Examples

```
# specify the path to GEDI level2A data (zip file)
outdir <- tempdir()
level2A_fp_zip <- system.file("extdata",
  "GEDI02_A_2019108080338_O01964_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
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 = mySetOfMetrics(rh100), res = 0.0005)
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 stars raster layer

Value

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

See Also

https://lpdaac.usgs.gov/products/gedi02_bv002/

Examples

```
# specify the path to GEDI level2B data (zip file)
outdir <- tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019108080338_O01964_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)
```

---

plot,gedi.fullwaveform,missing-method

Plot GEDI object*

Description

For gedi.fullwaveform: will plot the full waveform

Usage

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

Arguments

x	An object of class gedi.fullwaveform (output of getLevel1BWF() function)
relative	if TRUE, the Waveform Amplitude will be showed in percentage (%)
polygon	if TRUE, the polygon will be added to the plot
...	will be passed to the main plot

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_O01964_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
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)"
)

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

par(oldpar)
close(level1b)
```

`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.

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 ggplot2::ggplot package.

See Also

https://lpdaac.usgs.gov/products/gedi02_bv002/

Examples

```
# specify the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
                  "GEDI02_B_2019108080338_O01964_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)
```

plotPAVDProfile *Plot GEDI Plant Area Volume Density Profile*

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 ggplot2::ggplot package.

See Also

`https://lpdaac.usgs.gov/products/gedi02_bv002/`

Examples

```
# specify the path to GEDI level2B data (zip file)
outdir = tempdir()
level2B_fp_zip <- system.file("extdata",
                  "GEDI02_B_2019108080338_O01964_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).
`...`	Will be passed to the main plot.

Value

Nothing

See Also

https://lpdaac.usgs.gov/products/gedi02_bv002/

Examples

```
# specify the path to GEDI level1B and Level2A data (zip file)
outdir <- tempdir()
level1B_fp_zip <- system.file("extdata",
  "GEDI01_B_2019108080338_O01964_T05337_02_003_01_sub.zip",
  package = "rGEDI"
)

level2A_fp_zip <- system.file("extdata",
  "GEDI02_A_2019108080338_O01964_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)
```

`polyStatsLevel2AM` *Compute descriptive statistics of GEDI Elevation and Height Metrics*

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

```
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

Value

Returns an S4 object of class data.table::data.table Containing Statistics of GEDI level2A defined metrics

See Also

https://lpdaac.usgs.gov/products/gedi02_av002/

Examples

```
# Specifying the path to GEDI level2A data (zip file)
outdir <- tempdir()
level2A_fp_zip <- system.file("extdata",
  "GEDI02_A_2019108080338_O01964_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 sf object
library(sf)
polygon <- sf::st_read(polygon_filepath)

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

# Clipping GEDI data by geometry
level2AM_clip <- clipLevel2AMGeometry(level2AM, polygon, 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 = mySetOfMetrics(rh100),
  id = level2AM_clip$id
)

head(RH100metrics)

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::data.table Containing Statistics of GEDI level2BVPM defined metrics

See Also

`https://lpdaac.usgs.gov/products/gedi02_bv002/`

Examples

```
# Specifying the path to GEDI level2B data (zip file)
outdir <- tempdir()
level2B_fp_zip <- system.file("extdata",
  "GEDI02_B_2019108080338_O01964_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 sf object
library(sf)
polygon <- sf::st_read(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, 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_bv002/`

Examples

```
# Specifying the path to GEDI level1B data (zip file)
outdir = tempdir()
level1B_fp_zip <- system.file("extdata",
                  "GEDI01_B_2019108080338_O01964_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)

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_av002/

Examples

```
# Specifying the path to GEDI level2A data (zip file)
outdir = tempdir()
level2A_fp_zip <- system.file("extdata",
                  "GEDI02_A_2019108080338_O01964_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

```
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_O01964_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