Package 'rGEDI'

November 22, 2026

Type Package

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

Version 0.6.1

Description Provides an end-to-end workflow for NASA Global Ecosystem Dynamics Investigation (GEDI) Level 1B, Level 2A, Level 2B, Level 3, Level 4A, and Level 4B data. Functions search NASA catalogues, download granules, stream HDF5 products from Earthdata Cloud, and extract geolocations, waveforms, elevation and relative-height metrics, canopy cover, plant area profiles, and aboveground biomass. Methods quality-filter and spatially clip HDF5, tabular, and raster products; calculate footprint and polygon statistics; create gridded summaries; and visualize orbital tracks and measurements. The package also simulates GEDI-like waveforms from airborne laser-scanning point clouds and supports spatial sampling, predictive modelling, ancillary-data extraction, and wall-to-wall upscaling with optional Google Earth Engine workflows. An optional Python interface queries analysis-ready footprint data from the public 'gediDB' 'TileDB' service.

License GPL-3

Encoding UTF-8

Depends methods

Imports bit64, curl, data.table, fs, getPass, ggplot2, hdf5r, jsonlite, lazyeval, parallel, RColorBrewer, R6, sf, stats, terra

Suggests gifski, httr2, lattice, leaflet, leafsync, lidR, maps, plot3D, png, rasterVis, randomForest, reticulate, stringr, testthat (>= 3.0.0), viridis

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

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

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Carlos Alberto Silva [aut, cph, cre],
Caio Hamamura [aut, cph]

Maintainer Carlos Alberto Silva <c.silva@ufl.edu>

Repository CRAN

Date/Publication 2026-11-22 00:00:02 UTC

Contents

rGEDI-package 4
accumulateGEDIGranule 5
accumulateGEDIGranules 5
addEEImage 6
aggregateLevel3 7
aggregateLevel4B 7
aspect 8
assembleGEDIArchiveYears 8
build_ee_forest 9
clip 9
clipLevel1B 11
clipLevel1BGeo 13
clipLevel1BGeoGeometry 14
clipLevel1BGeometry 16
clipLevel2A 17
clipLevel2AGeometry 18
clipLevel2AM 20
clipLevel2AMGeometry 21
clipLevel2B 22
clipLevel2BGeometry 24
clipLevel2BPAIProfile 25
clipLevel2BPAIProfileGeometry 27
clipLevel2BPAVDProfile 28
clipLevel2BPAVDProfileGeometry 29
clipLevel2BVPM 31
clipLevel2BVPMGeometry 32
clipLevel3 34
clipLevel4A 34
clipLevel4AGeometry 35
clipLevel4B 36
close,gedi.level1b-method 36
close.gedi.cloud_h5 37
earthdata_login 37
earthdata_s3_credentials 38
ee_build_AlphaEarth_embedding_terrain_stack 38
ee_build_hls_s1c_terrain_stack 39
ee_check_task_status 39
ee_image_to_drive 40
ee_initialize 41

ee_rect_to_sf . . . 41
extractEE . . . 42
ext_to_ee . . . 42
fit_metrics . . . 43
fit_model . . . 43
gedi.fullwaveform-class . . . 44
gedi.level1b-class . . . 44
gedi.level2a-class . . . 45
gedi.level2b-class . . . 45
gedi.level4a-class . . . 45
gedidb_available . . . 46
gedidb_disconnect . . . 46
gediDownload . . . 47
gediEE . . . 48
gedifinder . . . 49
gediWFMetrics . . . 50
gediWFSimulator . . . 51
gee_available . . . 53
geomSampling . . . 54
getLevel1BGeo . . . 54
getLevel1BWF . . . 58
getLevel2AM . . . 59
getLevel2BPAIProfile . . . 61
getLevel2BPAVDProfile . . . 62
getLevel2BVPM . . . 64
getLevel3 . . . 66
getLevel4A . . . 67
getLevel4B . . . 68
getTileUrl . . . 69
getTrack . . . 69
get_catalog_id . . . 70
glcmTexture . . . 71
gridSampling . . . 71
gridStatsLevel2AM . . . 72
gridStatsLevel2BVPM . . . 73
gridStatsLevel4A . . . 75
map_create . . . 75
map_download . . . 76
map_view . . . 76
mosaicLevel3 . . . 77
mosaicLevel4B . . . 77
plot,gedi.fullwaveform,missing-method . . . 78
plot.gedi_var_selection . . . 79
plotLevel3 . . . 80
plotLevel4A . . . 80
plotLevel4B . . . 81
plotPAIProfile . . . 81
plotPAVDProfile . . . 83

plotWFMetrics . . . 84
plot_gedi_orbit_animation . . . 85
polyStats . . . 87
polyStatsLevel2AM . . . 87
polyStatsLevel2BVPM . . . 89
polyStatsLevel4A . . . 90
predict.gedi_model . . . 91
predictGEDIH5 . . . 91
projectLevel3 . . . 92
projectLevel4B . . . 93
randomSampling . . . 93
rasterize . . . 94
rasterSampling . . . 94
readGEDIArchiveRegion . . . 95
readLevel1B . . . 95
readLevel2A . . . 96
readLevel2B . . . 97
readLevel3 . . . 98
readLevel4A . . . 98
readLevel4B . . . 99
resampleLevel3 . . . 99
resampleLevel4B . . . 100
rGEDI_configure . . . 100
sample . . . 101
search_datasets . . . 101
slope . . . 102
spacedSampling . . . 102
stratifiedSampling . . . 103
to_vect . . . 103
varSel . . . 104
vect_as_ee . . . 104

Index **106**

`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/dataproducts/products/.

Author(s)

Carlos Alberto Silva and Caio Hamamura

See Also

For comprehensive examples refer to https://github.com/carlos-alberto-silva/rGEDI/blob/master/README.md

---

accumulateGEDIGranule *Accumulate the screened footprints of a cloud granule on a 1 km grid*

Description

Accumulate the screened footprints of a cloud granule on a 1 km grid

Usage

```
accumulateGEDIGranule(entry, from = NULL, to = NULL, region = NULL)
```

Arguments

entry	One manifest row with product, version, metric and url.
from, to	Optional UTC date bounds. The upper bound is exclusive.
region	Optional output of readGEDIArchiveRegion().

Value

A data.table of metric sums and footprint counts by EASE-Grid cell.

---

accumulateGEDIGranules *Accumulate multiple cloud granules on a common 1 km grid*

Description

Each worker opens and closes one cloud granule independently. Partial cell totals are combined after all workers finish, so the result is identical to summing sequential granule results.

Usage

```
accumulateGEDIGranules(
  entries,
  from = NULL,
  to = NULL,
  region = NULL,
  parallel = FALSE,
  workers = NULL
)
```

Arguments

`entries`	A data frame or data table of manifest rows accepted by `accumulateGEDIGranule()`.
`from, to`	Optional UTC date bounds. The upper bound is exclusive.
`region`	Optional output of `readGEDIArchiveRegion()`.
`parallel`	Logical; process granules concurrently.
`workers`	Number of workers. `NULL` uses physical cores minus one, limited by the number of granules.

Value

A data table of combined metric sums and footprint counts by EASE-Grid cell. The `workers` attribute records actual concurrency.

`addEEImage` *Add an Earth Engine image to a Leaflet map*

Description

Add an Earth Engine image to a Leaflet map

Usage

```
addEEImage(
  map,
  image,
  bands = NULL,
  min = 0,
  max = 1,
  palette = NULL,
  group = NULL,
  ...
)
```

Arguments

`map`	A Leaflet widget.
`image`	Earth Engine image.
`bands`	One or three bands.
`min, max`	Visualization limits.
`palette`	Optional palette.
`group`	Layer group.
`...`	Passed to `leaflet::addTiles()`.

Value

A Leaflet widget.

aggregateLevel3 *Aggregate a GEDI Level 3 raster*

Description

Aggregate a GEDI Level 3 raster

Usage

```
aggregateLevel3(x, fact, fun = mean, ...)
```

Arguments

x	A raster path or SpatRaster.
fact	Aggregation factor.
fun	Aggregation function.
...	Passed to terra::aggregate().

Value

A terra::SpatRaster.

aggregateLevel4B *Aggregate a GEDI Level 4B raster*

Description

Aggregate a GEDI Level 4B raster

Usage

```
aggregateLevel4B(x, fact, fun = mean, ...)
```

Arguments

x	A raster path or SpatRaster.
fact	Aggregation factor.
fun	Aggregation function.
...	Passed to terra::aggregate().

Value

A terra::SpatRaster.

`aspect` *Earth Engine terrain aspect*

Description

Earth Engine terrain aspect

Usage

```
aspect(x)
```

Arguments

`x`	Earth Engine elevation image.

Value

An Earth Engine image.

`assembleGEDIArchiveYears` *Render complete calendar years and their cumulative footprint grids*

Description

Render complete calendar years and their cumulative footprint grids

Usage

```
assembleGEDIArchiveYears(root, batch_size = 16L)
```

Arguments

`root`	Archive directory containing its index, manifests, markers and partials.
`batch_size`	Batch size used during the cloud pass.

Value

A list with the archive readiness flag and a coverage data.table.

build_ee_forest *Train a regression forest in Earth Engine*

Description

Train a regression forest in Earth Engine

Usage

```
build_ee_forest(training, response, predictors, trees = 500L, seed = 1L)
```

Arguments

`training`	Earth Engine `FeatureCollection` containing predictors and the response.
`response`	Response property name, such as `"agbd"` or `"rh98"`.
`predictors`	Predictor property names.
`trees`	Number of trees.
`seed`	Random seed.

Value

A trained Earth Engine classifier configured for regression.

clip *Clip GEDI data with a common interface*

Description

`clip()` provides one clipping interface for open GEDI HDF5 products, extracted footprint tables, and GEDI Level 3 or Level 4B rasters. It follows the workflow used by ICESat2VegR while retaining rGEDI's coordinate order.

Usage

```
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level1b,numeric'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level1b,SpatExtent'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level1b,ANY'
clip(x, clip_obj, ...)
```

```
## S4 method for signature 'gedi.level2a,numeric'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level2a,SpatExtent'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level2a,ANY'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level2b,numeric'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level2b,SpatExtent'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level2b,ANY'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level4a,numeric'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level4a,SpatExtent'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi.level4a,ANY'
clip(x, clip_obj, ...)

## S4 method for signature 'data.frame,numeric'
clip(x, clip_obj, ...)

## S4 method for signature 'data.frame,SpatExtent'
clip(x, clip_obj, ...)

## S4 method for signature 'data.frame,ANY'
clip(x, clip_obj, ...)

## S4 method for signature 'SpatRaster,ANY'
clip(x, clip_obj, ...)

## S4 method for signature 'gedi_gee_object,ANY'
clip(x, clip_obj, ...)
```

Arguments

x	A gedi.level1b, gedi.level2a, gedi.level2b, or gedi.level4a object; a footprint data.frame/data.table; or a terra::SpatRaster, or a lazy gedi_gee_object.
clip_obj	A numeric extent in the order c(xmin, xmax, ymin, ymax), a terra::SpatExtent,

	or an `sf`, `sfc`, or `terra::SpatVector` polygon object.
`...`	Arguments used by the selected method. HDF5 methods accept `output` and geometry methods also accept `split_by`. Table geometry methods accept `split_by`. Raster methods accept `filename` and `overwrite`.

Details

Extracted tables are recognized from standard GEDI coordinates: `longitude_bin0`/`latitude_bin0`, `lon_lowestmode`/`lat_lowestmode`, or `longitude`/`latitude`. When the corresponding `longitude_lastbin` and `latitude_lastbin` fields exist, both waveform endpoints must fall inside a numeric extent.

This function masks `graphics::clip()`. Use `graphics::clip()` explicitly when changing a base graphics clipping region.

Value

The return type follows `x`. Open Level 1B, 2A, 2B, and 4A HDF5 inputs return clipped open GEDI objects; footprint tables return a data.table::data.table; and Level 3/4B rasters return a `terra::SpatRaster`. Geometry clipping with `split_by` returns a named list of HDF5-backed GEDI objects, one per polygon ID. A GEE input returns a lazy GEE object of the same server-side type.

Examples

```
footprints <- data.frame(
  shot_number = 1:3,
  lon_lowestmode = c(-44.13, -44.12, -44.10),
  lat_lowestmode = c(-13.74, -13.73, -13.70)
)
clip(footprints, c(-44.14, -44.11, -13.75, -13.72))
```

`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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi01-b-002

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi01-b-002

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)

if (interactive() && requireNamespace("leaflet", quietly = TRUE)) {
  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 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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi01-b-002

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, beams = "BEAM0001")

# 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")

if (interactive() && requireNamespace("leaflet", quietly = TRUE)) {
  library(leaflet)
  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)
polygon <- polygon[1, , drop = FALSE]

# Spepecifing 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
)

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-a-002

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-a-002

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)
polygon <- polygon[1, , drop = FALSE]

# 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
)
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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-a-002

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, beams = "BEAM0001")

# 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, beams = "BEAM0001")

# 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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi01-b-002

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi01-b-002

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)
polygon <- polygon[1, , drop = FALSE]

# 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
)

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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
beam <- grep("^BEAM", (level2b@h5)$ls(recursive = FALSE)$name,
             value = TRUE)[1]
level2BVPM <- getLevel2BVPM(level2b, beams = beam)

# 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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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
beam <- grep("^BEAM", (level2b@h5)$ls(recursive = FALSE)$name,
             value = TRUE)[1]
level2BVPM <- getLevel2BVPM(level2b, beams = beam)

# 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)
```

---

clipLevel3 *Clip a GEDI Level 3 raster*

Description

Clip a GEDI Level 3 raster

Usage

```
clipLevel3(x, y, filename = "", overwrite = FALSE)
```

Arguments

x	A raster path or SpatRaster.
y	An extent c(xmin, xmax, ymin, ymax) or polygon object.
filename	Optional output filename.
overwrite	Logical.

Value

A terra::SpatRaster.

---

clipLevel4A *Clip GEDI Level 4A data by an extent*

Description

Clip GEDI Level 4A data by an extent

Usage

```
clipLevel4A(level4A, xmin, xmax, ymin, ymax, output = "")
```

Arguments

level4A	A gedi.level4a object returned by readLevel4A() or a table returned by getLevel4A().
xmin, xmax, ymin, ymax	Bounding coordinates in longitude/latitude.
output	Optional output HDF5 path when level4A is a gedi.level4a object. A temporary file is used by default.

Value

A clipped gedi.level4a object for HDF5 input, or a data.table::data.table for table input.

---

clipLevel4AGeometry *Clip GEDI Level 4A data by polygons*

Description

Clip GEDI Level 4A data by polygons

Usage

```
clipLevel4AGeometry(level4A, polygon, split_by = NULL, output = "")
```

Arguments

level4A	A gedi.level4a object returned by readLevel4A() or a table returned by getLevel4A().
polygon	An sf, sfc, or SpatVector polygon object.
split_by	Optional polygon attribute. For HDF5 input, one clipped file is returned per unique value. For table input, the value is copied to poly_id.
output	Optional output HDF5 path when level4A is a gedi.level4a object. A temporary file is used by default.

Value

A clipped gedi.level4a object (or a named list when split_by is used) for HDF5 input, or a data.table::data.table for table input.

clipLevel4B *Clip a GEDI Level 4B raster*

Description

Clip a GEDI Level 4B raster

Usage

```
clipLevel4B(x, y, filename = "", overwrite = FALSE)
```

Arguments

x	A raster path or SpatRaster.
y	An extent c(xmin, xmax, ymin, ymax) or polygon object.
filename	Optional output filename.
overwrite	Logical.

Value

A terra::SpatRaster.

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, ...)

## S4 method for signature 'gedi.level4a'
close(con, ...)
```

Arguments

con	An object of class gedi.level2b
...	Inherited from base

Value

No return value, called for the side effect of closing the underlying HDF5 connection.

---

close.gedi.cloud_h5 *Close a cloud HDF5 connection*

Description

Close a cloud HDF5 connection

Usage

```
## S3 method for class 'gedi.cloud_h5'
close(con, ...)
```

Arguments

con	A cloud-backed GEDI HDF5 object returned by a readLevel*() function.
...	Reserved for compatibility with close().

Value

Invisibly returns NULL.

---

earthdata_login *Authenticate with NASA Earthdata*

Description

Uses an existing bearer token, environment credentials, or .netrc file.

Usage

```
earthdata_login(persist = TRUE, netrc = Sys.getenv("NETRC", unset = ""))
```

Arguments

persist	Logical; allow Earthaccess to persist an interactive login.
netrc	Path to an Earthdata .netrc file.

Value

Invisibly returns TRUE after successful authentication.

earthdata_s3_credentials

Configure temporary NASA Earthdata S3 credentials

Description

Requests short-lived AWS credentials from Earthaccess and exports them for GDAL/terra. Direct S3 access to GEDI is available from AWS `us-west-2`.

Usage

```
earthdata_s3_credentials(daac = "ORNL")
```

Arguments

`daac`	NASA DAAC short name. GEDI products are hosted by ORNL DAAC.

Value

Invisibly returns the credential list.

ee_build_AlphaEarth_embedding_terrain_stack

Build an AlphaEarth and terrain predictor stack

Description

Build an AlphaEarth and terrain predictor stack

Usage

```
ee_build_AlphaEarth_embedding_terrain_stack(
  geom,
  start_year,
  end_year,
  add_lonlat = TRUE
)
```

Arguments

`geom`	Area of interest.
`start_year, end_year`	Inclusive year range.
`add_lonlat`	Add coordinate bands.

Value

An Earth Engine image.

`ee_build_hls_s1c_terrain_stack` *Build an HLS and terrain predictor stack in Earth Engine*

Description

Build an HLS and terrain predictor stack in Earth Engine

Usage

```
ee_build_hls_s1c_terrain_stack(
  x,
  start_date,
  end_date,
  cloud_max = 20,
  buffer_m = 30
)
```

Arguments

`x`	Area of interest.
`start_date, end_date`	Date range.
`cloud_max`	Maximum HLS cloud cover.
`buffer_m`	AOI buffer in meters.

Value

An Earth Engine image.

`ee_check_task_status` *Check an Earth Engine task*

Description

Check an Earth Engine task

Usage

```
ee_check_task_status(task, quiet = TRUE)
```

Arguments

task	Earth Engine batch task.
quiet	Logical.

Value

Task status list.

---

ee_image_to_drive *Export an Earth Engine image to Google Drive*

Description

Creates a Google Earth Engine batch export task for a GeoTIFF in Google Drive. This is useful for images that exceed the direct download limit.

Usage

```
ee_image_to_drive(
  image,
  description = "rGEDI_export",
  folder = "EE_Exports",
  file_name_prefix = description,
  region,
  scale = 30,
  crs = "EPSG:4326",
  max_pixels = 1e+13,
  start = TRUE,
  ...
)
```

Arguments

image	Earth Engine image.
description	Task description shown in Earth Engine.
folder	Google Drive folder.
file_name_prefix	Output file prefix.
region	Area of interest accepted by ext_to_ee().
scale	Pixel size in meters.
crs	Output CRS.
max_pixels	Maximum number of pixels allowed by Earth Engine.
start	Start the task immediately.
...	Additional arguments passed to Earth Engine's toDrive method.

Value

An Earth Engine batch task. Use `ee_check_task_status()` to monitor it.

---

ee_initialize *Initialize Google Earth Engine*

Description

Initialize Google Earth Engine

Usage

```
ee_initialize(
  project = Sys.getenv("EE_PROJECT", unset = ""),
  authenticate = TRUE,
  quiet = FALSE
)
```

Arguments

`project`	Google Cloud project ID.
`authenticate`	Authenticate when initialization fails.
`quiet`	Logical.

Value

Invisibly returns the Python Earth Engine module.

---

ee_rect_to_sf *Convert an Earth Engine rectangle to* `sf`

Description

Convert an Earth Engine rectangle to `sf`

Usage

```
ee_rect_to_sf(aoi)
```

Arguments

`aoi`	Earth Engine geometry.

Value

An `sf` polygon.

extractEE *Extract Earth Engine image values at GEDI footprints*

Description

Extract Earth Engine image values at GEDI footprints

Usage

```
extractEE(stack, geom, scale = 30, chunk_size = 1000L)

extractGEDIAncillary(stack, geom, scale = 30, chunk_size = 1000L)
```

Arguments

stack	Earth Engine image or list of images.
geom	GEDI table, sf, or SpatVector points.
scale	Sampling scale in meters.
chunk_size	Number of points per request.

Value

A data.table::data.table.

ext_to_ee *Convert an extent to an Earth Engine geometry*

Description

Convert an extent to an Earth Engine geometry

Usage

```
ext_to_ee(x)
```

Arguments

x	A SpatExtent, raster/vector object, or numeric extent in c(xmin, xmax, ymin, ymax) order.

Value

An Earth Engine geometry.

fit_metrics *Model fit metrics*

Description

Model fit metrics

Usage

```
fit_metrics(observed, predicted, plotstat = FALSE, unit = "", ...)
```

Arguments

observed, predicted	Numeric vectors.
plotstat	Draw an observed-versus-predicted plot.
unit	Unit label.
...	Passed to graphics::plot().

Value

A data frame of accuracy statistics.

fit_model *Fit a GEDI statistical or random-forest model*

Description

Fit a GEDI statistical or random-forest model

Usage

```
fit_model(
  x,
  y,
  method = c("randomForest", "lm"),
  test = c("none", "split", "kfold", "loocv", "bootstrap"),
  k = 5L,
  test_size = 0.3,
  iterations = 100L,
  seed = NULL,
  ...
)
```

Arguments

x	Predictor data frame.
y	Response vector or response column name in x.
method	"randomForest" or "lm".
test	Validation method: "none", "split", "kfold", "loocv", or "bootstrap".
k	Number of folds.
test_size	Fraction held out for split validation.
iterations	Bootstrap iterations.
seed	Optional random seed.
...	Model arguments.

Value

An object of class gedi_model.

---

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 A local H5File or a cloud-backed GEDI HDF5 connection containing geolocated waveforms.

See Also

H5File in the hdf5r package and https://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi01-b-002

gedi.level2a-class *Class for GEDI level2A*

Description

Class for GEDI level2A

Slots

`h5` A local `H5File` or cloud-backed GEDI HDF5 connection containing ground elevation, canopy top height, and relative heights (RH).

See Also

`H5File` in the `hdf5r` package and https://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-a-002

gedi.level2b-class *Class for GEDI level2B*

Description

Class for GEDI level2B

Slots

`h5` A local `H5File` or cloud-backed GEDI HDF5 connection containing canopy cover, PAI, PAVD, and FHD.

See Also

`H5File` in the `hdf5r` package and https://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

gedi.level4a-class *Class for GEDI Level 4A footprint biomass data*

Description

Class for GEDI Level 4A footprint biomass data

Slots

`h5` An open local or cloud-backed HDF5 connection containing a GEDI04_A granule.

gedidb_available *Test availability of the public gediDB service*

Description

Checks that the optional Python module is installed and that the selected public GFZ gediDB bucket is reachable. The default check is deliberately lightweight. Set `deep = TRUE` to open the TileDB array and read its variable metadata; that check can take substantially longer.

Usage

```
gedidb_available(version = "003", deep = FALSE, timeout = 5)
```

Arguments

`version`	GEDI database release, `"003"` or `"002"`.
`deep`	Logical; also open the TileDB array and read its metadata.
`timeout`	Timeout in seconds for the lightweight endpoint request.

Value

A single logical value. A diagnostic explanation is stored in its `reason` attribute.

gedidb_disconnect *Close cached gediDB connections*

Description

Releases the cached Python TileDB providers used to make repeated queries fast. New providers are created automatically by the next query.

Usage

```
gedidb_disconnect()
```

Value

No return value, called for the side effect of closing cached providers.

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,
  netrc = Sys.getenv("NETRC", unset = "")
)
```

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.
netrc	Optional path to an existing NASA Earthdata .netrc file. By default, uses the NETRC environment variable before prompting.

Value

The normalized paths of the downloaded files, invisibly. On failure the function stops with an error.

References

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

Examples

```
# 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()

if (nzchar(Sys.getenv("NASA_USER")) && nzchar(Sys.getenv("NASA_PASSWORD"))) {
# 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)
}
```

---

gediEE *Open a GEDI product in Google Earth Engine*

Description

Open a GEDI product in Google Earth Engine

Usage

```
gediEE(
  product = names(.gedi_ee_catalog),
  start_date = NULL,
  end_date = NULL,
  aoi = NULL,
  quality = TRUE,
  max_granules = 200L
)
```

Arguments

product	GEDI Earth Engine product name.
start_date, end_date	Optional dates for image collections.

aoi	Optional spatial filter.
quality	Apply available quality masks.
max_granules	Maximum number of vector granules to merge. Vector GEDI products are stored as folders of per-granule feature collections in Earth Engine, so use dates and/or an AOI to keep the request focused.

Value

An Earth Engine image, image collection, or feature collection.

`gedifinder` *Find GEDI granules through NASA CMR*

Description

Find GEDI granules through NASA CMR

Usage

```
gedifinder(
  product,
  ul_lat = NULL,
  ul_lon = NULL,
  lr_lat = NULL,
  lr_lon = NULL,
  version = NULL,
  daterange = NULL,
  access = c("https", "s3", "opendap", "all"),
  cloud_hosted = TRUE,
  return = c("url", "table"),
  page_size = 2000L,
  cloud_computing = FALSE,
  persist = TRUE,
  orbit = NULL,
  source = c("earthdata", "gedidb", "gee"),
  check = TRUE
)
```

Arguments

product	One of `GEDI01_B`, `GEDI02_A`, `GEDI02_B`, `GEDI03`, `GEDI04_A`, or `GEDI04_B`.
ul_lat, ul_lon, lr_lat, lr_lon	Bounding coordinates in decimal degrees.
version	Product version. `NULL` selects the current supported version.
daterange	Optional two-element date or date-time vector.
access	Requested link type: `"https"`, `"s3"`, `"opendap"`, or `"all"`.

cloud_hosted	Logical retained for backward compatibility. The current GEDI collection identifiers resolve to their Earthdata Cloud holdings.
return	Return a URL vector or a metadata table.
page_size	CMR page size.
cloud_computing	Logical. When TRUE, authenticate with Earthdata and return HTTPS granules marked for direct cloud streaming by the product-specific readLevel*() function.
persist	Logical; allow Earthaccess to persist an interactive login when cloud_computing = TRUE.
orbit	Optional GEDI orbit number, such as 1964 or "001964". When supplied, the spatial bounding box is omitted so CMR returns every production granule belonging to that orbit.
source	Data service. "earthdata" searches NASA CMR, "gedidb" returns a lazy reference to the public GFZ TileDB database, and "gee" returns a lazy Google Earth Engine asset reference.
check	Logical; for source = "gedidb" or "gee", verify that the optional backend and service are available before returning.

Value

For Earthdata, a character vector or data.table::data.table. For gediDB with return = "url", a gedidb_reference containing the remote address and search bounds; pass it to the corresponding readLevel*() function. A GEE URL result is a gedi_gee_reference that can be passed to the corresponding reader. With return = "table", either remote source returns one metadata row.

See Also

https://cmr.earthdata.nasa.gov/search/site/docs/search/api.html

---

gediWFMetrics *Compute metrics from GEDI or simulated full waveforms*

Description

Compute metrics from GEDI or simulated full waveforms

Usage

```
gediWFMetrics(
  input,
  outRoot = NULL,
  rhRes = 5,
  ground = TRUE,
  rhoG = 0.4,
```

```
  rhoC = 0.57,
  ground_method = c("auto", "truth", "lowest_peak", "quantile"),
  ...
)
```

Arguments

input	A gedi.level1b object or list of such objects.
outRoot	Optional output filename root.
rhRes	Relative-height percentile spacing.
ground	Logical; include the estimated ground elevation.
rhoG, rhoC	Ground and canopy reflectance.
ground_method	Ground estimator. "auto" uses the classified ground waveform when present and otherwise the lowest smoothed waveform peak.
...	Compatibility arguments from the original native implementation.

Value

A data.table::data.table with waveform metrics.

---

gediWFSimulator *Simulate GEDI-like full waveforms from airborne laser scanning data*

Description

This portable implementation follows the core gediRat algorithm: Gaussian footprint weighting, optional return-density normalization, vertical binning, pulse convolution, separation of LAS class 2 ground returns, and unit-integral normalization. Vectorized binning replaces the original point loop while avoiding its native GSL/GDAL build stack.

Usage

```
gediWFSimulator(
  input,
  output = tempfile(fileext = ".h5"),
  ground = TRUE,
  ascii = FALSE,
  waveID = NULL,
  coords = NULL,
  listCoord = NULL,
  gridBound = NULL,
  gridStep = 30,
  pSigma = -1,
  pFWHM = 15,
  pRes = NULL,
```

```
  fSigma = 5.5,
  res = 0.15,
  decimate = 1,
  maxBins = 1024L,
  countOnly = FALSE,
  method = c("count", "intensity", "fraction"),
  normalize = TRUE,
  density_res = 1.5,
  intensity_threshold = 6e-04,
  noise = 0,
  keepOld = FALSE,
  seed = NULL,
  parallel = FALSE,
  workers = NULL,
  ...
)
```

Arguments

input	LAS/LAZ paths, a lidR::LAS, SpatVector, or table with X/Y/Z.
output	Output HDF5 path.
ground	Retained for compatibility; ground returns remain in the waveform.
ascii	Return/write a simple ASCII waveform instead of HDF5.
waveID	Optional waveform identifiers.
coords	A coordinate pair or matrix of footprint centers.
listCoord	Text file containing footprint centers.
gridBound	Optional c(xmin, xmax, ymin, ymax) simulation grid.
gridStep	Grid spacing in input coordinate units.
pSigma	Pulse sigma in meters. A negative value derives it from pFWHM.
pFWHM	Pulse full width at half maximum in nanoseconds.
pRes	Pulse sampling resolution. The default uses res, matching gediRat unless its -pRes option is supplied.
fSigma	Footprint sigma in input coordinate units.
res	Vertical waveform resolution.
decimate	Probability of retaining an ALS return.
maxBins	Maximum number of waveform bins.
countOnly	Use count-based returns rather than ALS intensity. Gaussian footprint weights are still applied.
method	Waveform contribution: "count", "intensity", or "fraction". countOnly = TRUE selects "count" for compatibility.
normalize	Correct for local ALS sampling density, as in the original simulator's default behavior.
density_res	Cell size used for sampling-density normalization.

`intensity_threshold`
: Smallest retained Gaussian footprint weight.

`noise`
: Relative Gaussian noise standard deviation. For example, `noise = 0.03` adds noise with a standard deviation equal to three percent of the waveform maximum. Zero leaves the waveform unchanged.

`keepOld`
: Do not overwrite an existing output file.

`seed`
: Optional random seed.

`parallel`
: Logical; simulate independent footprint centres concurrently.

`workers`
: Number of parallel workers. `NULL` uses physical cores minus one, limited by the number of footprints.

`...`
: Compatibility arguments accepted from the original simulator.

Value

A `gedi.level1b` object, or a waveform table when `ascii=TRUE`.

References

Hancock et al. (2019) doi:10.1029/2018EA000506

`gee_available` *Test Google Earth Engine availability*

Description

Verifies that the optional Earth Engine Python API is installed, initialized, and can access the GEDI catalogue.

Usage

```
gee_available()
```

Value

A single logical value with a diagnostic reason attribute.

geomSampling *GEDI polygon-stratified sampling specification*

Description

GEDI polygon-stratified sampling specification

Usage

```
geomSampling(size, geom, split_id = NULL, lon = NULL, lat = NULL)
```

Arguments

size	Number or fraction per polygon.
geom	Polygon object.
split_id	Optional polygon ID column.
lon, lat	Optional coordinate column names.

Value

A gedi_sampling_method object containing the polygon-stratified sampling parameters for use with sample().

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 = c("elevation_bin0", "elevation_lastbin"),
  beams = NULL,
  parallel = FALSE,
  workers = NULL
)
```

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.
`beams`	Optional beam names. `NULL` reads every beam.
`parallel`	Logical; process a multi-granule collection concurrently.
`workers`	Number of parallel workers, or `NULL` for automatic selection.

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi01-b-002

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi01-b-002

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,
  beams = NULL,
  include_rh = TRUE,
  rh = NULL,
  quality = FALSE,
  collect = TRUE,
  max_features = 100000L,
  chunk_size = 5000L,
  max_granules = 200L,
  parallel = FALSE,
  workers = NULL
)
```

Arguments

level2a	A GEDI Level2A object (output of readLevel2A() function). An S4 object of class "gedi.level2a".
beams	Optional beam names. NULL reads every beam.
include_rh	Logical. Include relative-height columns. Set to FALSE for a faster metadata-only read.
rh	Optional RH percentiles from 0 through 100. NULL retains all RH metrics for HDF5 data and requests the full RH profile from gediDB.
quality	Logical; for gediDB, retain Release 3 quality observations.
collect	Logical; collect a remote footprint query into R. Set to FALSE to retain a lazy GEE or gediDB object for rasterization.
max_features	Maximum GEE footprints allowed when collecting.
chunk_size	Number of GEE features transferred per request.
max_granules	Maximum vector granules merged by GEE.
parallel	Logical; process a multi-granule collection concurrently.
workers	Number of parallel workers, or NULL for automatic selection.

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

A data.table::data.table containing elevation and relative-height metrics, or a lazy remote query when collect = FALSE.

See Also

https://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-a-002

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, beams = NULL)
```

Arguments

`level2b`	A GEDI Level2B object (output of `readLevel2B()` function). An S4 object of class "gedi.level2b".
`beams`	Optional beam names. `NULL` reads every beam.

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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, beams = NULL)
```

Arguments

level2b	A GEDI Level2B object (output of readLevel2B() function). An S4 object of class "gedi.level2b".
beams	Optional beam names. NULL reads every beam.

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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"),
  beams = NULL,
  quality = FALSE,
  collect = TRUE,
  max_features = 100000L,
  chunk_size = 5000L,
  max_granules = 200L,
  parallel = FALSE,
  workers = NULL
)
```

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.
beams	Optional beam names. NULL reads every beam.
quality	Logical; for gediDB, retain Release 3 quality observations.
collect	Logical; collect remote footprints into R or retain a lazy GEE/gediDB object for rasterization.
max_features	Maximum GEE footprints allowed when collecting.
chunk_size	Number of GEE features transferred per request.
max_granules	Maximum vector granules merged by GEE.
parallel	Logical; process a multi-granule collection concurrently.
workers	Number of parallel workers, or NULL for automatic selection.

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

A data.table::data.table containing the Vegetation Profile Biophysical Variables, or a lazy remote query when `collect = FALSE`.

See Also

https://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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)
```

`getLevel3` *Get GEDI Level 3 pixel measurements*

Description

Get GEDI Level 3 pixel measurements

Usage

```
getLevel3(
  x,
  geometry = NULL,
  metrics = NULL,
  xy = TRUE,
  cells = FALSE,
  na.rm = TRUE,
  global = FALSE,
  max_cells = 5e+06,
  ...
)
```

Arguments

`x`	A Level 3 raster returned by `readLevel3()` or a raster path.
`geometry`	Optional points, polygons, or extent accepted by `terra::vect()`. A geometry is required for Google Earth Engine data.
`metrics`	Optional raster layers to retrieve.
`xy`	Include cell-center coordinates when returning all cells.
`cells`	Include cell numbers when returning all cells.
`na.rm`	Remove rows whose raster values are all missing.
`global`	Explicitly allow conversion of a large complete raster.
`max_cells`	Maximum cells converted without `global = TRUE`.
`...`	Passed to `terra::extract()`, `terra::as.data.frame()`, or `extractEE()`.

Value

A data.table::data.table containing pixel measurements.

getLevel4A *Extract GEDI Level 4A footprint metrics*

Description

Extract GEDI Level 4A footprint metrics

Usage

```
getLevel4A(
  level4a,
  cols = .level4a_default_columns,
  quality = FALSE,
  beams = NULL,
  collect = TRUE,
  max_features = 100000L,
  chunk_size = 5000L,
  max_granules = 200L,
  parallel = FALSE,
  workers = NULL
)
```

Arguments

level4a	A gedi.level4a object returned by readLevel4A().
cols	Character vector of fields to return. NULL returns all datasets shared by the selected beams.
quality	Logical. If TRUE, retain observations accepted by the current Release 3 quality fields when present: • l4a_quality_flag_rel3 • degrade_include_flag • elev_highestreturn_outlier_flag Legacy Release 2 quality fields remain supported.
beams	Optional character vector of GEDI beam names.
collect	Logical; collect remote footprints into R or retain a lazy GEE/gediDB query for rasterization.
max_features	Maximum GEE footprints allowed when collecting.
chunk_size	Number of GEE features transferred per request.
max_granules	Maximum vector granules merged by GEE.
parallel	Logical; process a multi-granule collection concurrently.
workers	Number of parallel workers, or NULL for automatic selection.

Value

A data.table::data.table with one row per footprint, or a lazy remote query object when `collect = FALSE`.

`getLevel4B` *Get GEDI Level 4B pixel measurements*

Description

Get GEDI Level 4B pixel measurements

Usage

```
getLevel4B(
  x,
  geometry = NULL,
  metrics = NULL,
  xy = TRUE,
  cells = FALSE,
  na.rm = TRUE,
  global = FALSE,
  max_cells = 5e+06,
  ...
)
```

Arguments

`x`	A Level 4B raster returned by `readLevel4B()`, a raster path, or a lazy Google Earth Engine Level 4B object.
`geometry`	Optional points, polygons, or extent accepted by `terra::vect()`. A geometry is required for Google Earth Engine data.
`metrics`	Optional raster layers to retrieve.
`xy`	Include cell-center coordinates when returning all cells.
`cells`	Include cell numbers when returning all cells.
`na.rm`	Remove rows whose raster values are all missing.
`global`	Explicitly allow conversion of a large complete raster.
`max_cells`	Maximum cells converted without `global = TRUE`.
`...`	Passed to `terra::extract()`, `terra::as.data.frame()`, or `extractEE()`.

Value

A data.table::data.table containing pixel-level biomass measurements.

getTileUrl *Get an Earth Engine image tile URL*

Description

Get an Earth Engine image tile URL

Usage

```
getTileUrl(image, vis = list())
```

Arguments

image	Earth Engine image.
vis	Visualization parameter list.

Value

Character tile URL.

getTrack *Extract a GEDI reference ground track*

Description

Standardizes footprint coordinates, acquisition order, and beam identifiers from any open point-level GEDI product or an extracted footprint table. The result can be plotted directly or passed to `plot_gedi_orbit_animation()`.

Usage

```
getTrack(
  x,
  lon = NULL,
  lat = NULL,
  time = NULL,
  track = NULL,
  beams = NULL,
  every = 1L,
  segment_gaps = TRUE
)
```

Arguments

x	A data frame, data.table, sf or SpatVector containing GEDI footprint coordinates; an open gedi.level1b, gedi.level2a, gedi.level2b, or gedi.level4a object; or a list of these objects from consecutive production granules in one orbit.
lon, lat	Names of longitude and latitude columns. When NULL, common GEDI coordinate names are detected automatically.
time	Optional name of a column used to order footprints within each track. GEDI delta_time and shot_number columns are detected by default.
track	Optional name of a beam or track column. Separate tracks are colored independently.
beams	Optional beam names to read from an open GEDI object.
every	Keep every nth footprint after ordering. This is useful when plotting a complete orbit.
segment_gaps	Append segment numbers to track names when unusually large spatial gaps occur. Animation keeps this disabled so consecutive GEDI production granules remain one beam for a complete orbit.

Value

A data.table::data.table with longitude, latitude, sequence, track, and available time and shot identifiers.

Examples

```
shots <- data.frame(
  lon_lowestmode = seq(-44.2, -44.1, length.out = 10),
  lat_lowestmode = seq(-13.8, -13.7, length.out = 10),
  delta_time = seq_len(10), beam = "BEAM0000"
)
getTrack(shots, every = 2)
```

---

get_catalog_id *Return an Earth Engine catalog ID*

Description

Return an Earth Engine catalog ID

Usage

```
get_catalog_id(x)
```

Arguments

x	A GEDI product name, catalog search row, or catalog ID.

Value

A character catalog ID.

`glcmTexture` *Earth Engine gray-level co-occurrence textures*

Description

Earth Engine gray-level co-occurrence textures

Usage

```
glcmTexture(x, size = 1L, kernel = NULL, average = TRUE)
```

Arguments

`x`	Earth Engine integer image.
`size`	Neighborhood size.
`kernel`	Optional Earth Engine kernel.
`average`	Average directional metrics.

Value

An Earth Engine image.

`gridSampling` *GEDI grid-stratified sampling specification*

Description

GEDI grid-stratified sampling specification

Usage

```
gridSampling(size, grid_size, lon = NULL, lat = NULL)
```

Arguments

`size`	Number or fraction per cell.
`grid_size`	Grid size in decimal degrees.
`lon, lat`	Optional coordinate column names.

Value

A `gedi_sampling_method` object containing the grid-stratified sampling parameters for use with `sample()`.

`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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-a-002

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
)
if (interactive()) plot(ZTstats)

if (interactive()) {
  # Computing the maximum of RH100 only
  maxRH100 <- gridStatsLevel2AM(level2AM, mySetOfMetrics(rh100), res = 0.0005)
  plot(maxRH100)

  # Computing the mean of ZG only
  ZGmean <- gridStatsLevel2AM(level2AM, 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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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)
```

`gridStatsLevel4A` *Grid GEDI Level 4A footprint metrics*

Description

Grid GEDI Level 4A footprint metrics

Usage

```
gridStatsLevel4A(level4A, metric = "agbd", fun = mean, res = 0.01, ...)
```

Arguments

`level4A`	A table returned by `getLevel4A()`.
`metric`	Metric column name.
`fun`	Aggregation function.
`res`	Output resolution in decimal degrees.
`...`	Additional arguments passed to `fun`.

Value

A `terra::SpatRaster`.

`map_create` *Apply an Earth Engine model to a predictor image*

Description

Apply an Earth Engine model to a predictor image

Usage

```
map_create(model, stack, aoi = NULL, name = "prediction")
```

Arguments

`model`	A trained Earth Engine classifier or regressor.
`stack`	Earth Engine predictor image.
`aoi`	Optional area to clip.
`name`	Output band name.

Value

An Earth Engine image.

map_download *Download an Earth Engine image*

Description

Download an Earth Engine image

Usage

```
map_download(
  image,
  filename,
  region,
  scale = 30,
  crs = "EPSG:4326",
  overwrite = FALSE
)
```

Arguments

image	Earth Engine image.
filename	Destination GeoTIFF path.
region	Area of interest.
scale	Pixel size in meters.
crs	Output CRS.
overwrite	Logical.

Value

Invisibly returns filename.

map_view *Display Earth Engine layers in Leaflet*

Description

Display Earth Engine layers in Leaflet

Usage

```
map_view(layers, vis = list(), aoi = NULL)
```

Arguments

layers	Named list of Earth Engine images.
vis	Named list of visualization parameter lists.
aoi	Optional spatial object used to fit map bounds.

Value

A Leaflet widget.

---

mosaicLevel3 *Mosaic GEDI Level 3 rasters*

Description

Mosaic GEDI Level 3 rasters

Usage

```
mosaicLevel3(x, fun = "mean", filename = "", overwrite = FALSE)
```

Arguments

x	Raster paths, a SpatRaster, or a list of rasters.
fun	Function used for overlapping cells.
filename	Optional output filename.
overwrite	Logical.

Value

A terra::SpatRaster.

---

mosaicLevel4B *Mosaic GEDI Level 4B rasters*

Description

Mosaic GEDI Level 4B rasters

Usage

```
mosaicLevel4B(x, fun = "mean", filename = "", overwrite = FALSE)
```

Arguments

x	Raster paths, a SpatRaster, or a list of rasters.
fun	Function used for overlapping cells.
filename	Optional output filename.
overwrite	Logical.

Value

A terra::SpatRaster.

---

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)
```

`plot.gedi_var_selection` *Plot GEDI variable selection results*

Description

Plot GEDI variable selection results

Usage

```
## S3 method for class 'gedi_var_selection'
plot(x, which = c("importance", "rfe"), ...)
```

Arguments

`x`	A `gedi_var_selection` object.
`which`	Plot variable importance or RFE error.
`...`	Additional graphical parameters.

Value

The object, invisibly.

plotLevel3 *Plot GEDI Level 3 metrics*

Description

Plot GEDI Level 3 metrics

Usage

```
plotLevel3(x, ...)
```

Arguments

x	A raster path or SpatRaster.
...	Passed to terra::plot().

Value

Invisibly returns the raster.

plotLevel4A *Plot GEDI Level 4A footprint biomass*

Description

Plot GEDI Level 4A footprint biomass

Usage

```
plotLevel4A(level4A, metric = "agbd", ...)
```

Arguments

level4A	A table returned by getLevel4A().
metric	Metric column to display.
...	Additional arguments passed to graphics::plot().

Value

Invisibly returns level4A.

plotLevel4B *Plot GEDI Level 4B metrics*

Description

Plot GEDI Level 4B metrics

Usage

```
plotLevel4B(x, ...)
```

Arguments

x	A raster path or SpatRaster.
...	Passed to terra::plot().

Value

Invisibly returns the raster.

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

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)
```

---

plot_gedi_orbit_animation

Animate a GEDI ground track around the Earth

Description

Creates an interactive 3D HTML animation or an animated GIF from GEDI footprint coordinates. The presentation follows the ICESat2VegR orbit animation, with a rotating Earth, a NASA ISS illustration carrying GEDI, and a red near-infrared laser and ground track. ISS motion follows the time-ordered geolocation of the most complete beam extracted from the supplied HDF5 object.

Usage

```
plot_gedi_orbit_animation(
  x,
  lon = NULL,
  lat = NULL,
```

```
  time = NULL,
  track = NULL,
  output_file = tempfile("rGEDI-orbit-", fileext = ".html"),
  title = "GEDI orbit animation",
  duration = 18,
  launch = interactive(),
  track_speed = 2,
  earth_rotation_speed = 2,
  beams = NULL,
  every = 1L
)
```

Arguments

x	A data frame, `data.table`, `sf` or `SpatVector` containing GEDI footprint coordinates; an open `gedi.level1b`, `gedi.level2a`, `gedi.level2b`, or `gedi.level4a` object; or a list of these objects from consecutive production granules in one orbit.
lon, lat	Names of longitude and latitude columns. When `NULL`, common GEDI coordinate names are detected automatically.
time	Optional name of a column used to order footprints within each track. GEDI `delta_time` and `shot_number` columns are detected by default.
track	Optional name of a beam or track column. Separate tracks are colored independently.
output_file	Path of the `.html` or `.gif` file to create. GIF output requires the suggested `gifski` package.
title	Title shown above the animation.
duration	Duration of one animation cycle in seconds.
launch	Open the animation in the default browser after it is written.
track_speed	Initial HTML track playback speed from 1 to 15.
earth_rotation_speed	Initial Earth rotation speed from 0 to 20.
beams	Optional beam names to read from open GEDI objects.
every	Keep every nth footprint after ordering. Increase this for a complete orbit to reduce animation size and rendering time.

Value

The normalized path to the HTML file, invisibly.

Examples

```
orbit <- data.frame(
  longitude = seq(-60, -45, length.out = 30),
  latitude = seq(-8, 5, length.out = 30),
  delta_time = seq_len(30), beam = "BEAM0101"
```

```
)
html <- plot_gedi_orbit_animation(
  orbit, output_file = tempfile(fileext = ".html")
)
file.exists(html)
```

`polyStats` *Polygon summaries for any GEDI raster*

Description

Works with native Level 3 and Level 4B products and with rasters created from Level 2A, Level 2B, or Level 4A footprint measurements.

Usage

```
polyStats(x, polygon, fun = mean, na.rm = TRUE, ...)
```

Arguments

`x`	A raster path or terra::SpatRaster.
`polygon`	Polygon geometry accepted by `terra::vect()`.
`fun`	Summary function.
`na.rm`	Logical.
`...`	Passed to `terra::extract()`.

Value

A data.table::data.table with one row per polygon and summary columns for every raster layer.

`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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-a-002

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))

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

# Computing a serie statistics for GEDI metrics stratified by polygon
RH100metrics <- polyStatsLevel2AM(level2AM_clip,
  func = mySetOfMetrics(rh100),
  id = "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://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-b-002

Examples

```
# A small extracted Level 2B table
level2BVPM <- data.table::data.table(
  pai = c(1.1, 1.8, 2.2, 0.9, 1.5, 2.4),
  cover = c(0.31, 0.44, 0.52, 0.27, 0.39, 0.58),
  poly_id = rep(c("stand_1", "stand_2"), each = 3)
)
head(level2BVPM)

# 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, func = max(pai), id = NULL)
pai_max

# Computing the max of the Total Plant Area Index stratified by polygon
pai_max_poly <- polyStatsLevel2BVPM(level2BVPM, 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,
  func = mySetOfMetrics(cover),
  id = "poly_id"
)
head(cover_metrics)
```

`polyStatsLevel4A` *Polygon statistics for GEDI Level 4A footprints*

Description

Polygon statistics for GEDI Level 4A footprints

Usage

```
polyStatsLevel4A(level4A, polygon, metric = "agbd", fun = mean, id = NULL, ...)
```

Arguments

`level4A`	A table returned by `getLevel4A()`.

polygon	An sf, sfc, or SpatVector polygon object.
metric	Metric column name.
fun	Aggregation function.
id	Optional polygon ID field.
...	Additional arguments passed to fun.

Value

A data table with one row per polygon.

---

predict.gedi_model *Predict from a fitted GEDI model*

Description

Predict from a fitted GEDI model

Usage

```
## S3 method for class 'gedi_model'
predict(object, newdata, ...)
```

Arguments

object	A model returned by fit_model().
newdata	Predictor table containing the model's required columns.
...	Passed to the underlying fitted model's stats::predict() method.

Value

A numeric vector with one prediction per row of newdata.

---

predictGEDIH5 *Predict values and write them to HDF5*

Description

Predict values and write them to HDF5

Usage

```
predictGEDIH5(model, data, output, dataset = "prediction")
```

Arguments

model	A gedi_model or fitted R model.
data	Predictor table.
output	Output HDF5 filename.
dataset	Dataset name.

Value

Invisibly returns output.

---

projectLevel3 *Project a GEDI Level 3 raster*

Description

Project a GEDI Level 3 raster

Usage

```
projectLevel3(x, crs, method = "bilinear", ...)
```

Arguments

x	A raster path or SpatRaster.
crs	Target CRS.
method	Resampling method.
...	Passed to terra::project().

Value

A terra::SpatRaster.

projectLevel4B *Project a GEDI Level 4B raster*

Description

Project a GEDI Level 4B raster

Usage

```
projectLevel4B(x, crs, method = "bilinear", ...)
```

Arguments

x	A raster path or SpatRaster.
crs	Target CRS.
method	Resampling method.
...	Passed to terra::project().

Value

A terra::SpatRaster.

randomSampling *GEDI random sampling specification*

Description

GEDI random sampling specification

Usage

```
randomSampling(size)
```

Arguments

size	Number or fraction of observations to select.

Value

A gedi_sampling_method object containing the method name and requested sample size for use with sample().

rasterize *Rasterize GEDI measurements or standard terra objects*

Description

GEDI footprint tables and lazy gediDB or Google Earth Engine queries use rGEDI's metric-aware implementation. Other inputs are forwarded unchanged to `terra::rasterize()`.

Usage

```
rasterize(x, ...)
```

Arguments

`x`	A GEDI footprint table, lazy GEDI query, or an object supported by `terra::rasterize()`.
`...`	For GEDI inputs, supply `metric`, `res`, `fun`, optional `lon` and `lat`, `template`, `input_crs`, `output_crs`, parallel controls, and an optional `filename`. For other inputs these arguments are passed unchanged to `terra::rasterize()`.

Value

A terra::SpatRaster or lazy `gedi_gee_object`.

rasterSampling *GEDI raster-stratified sampling specification*

Description

GEDI raster-stratified sampling specification

Usage

```
rasterSampling(size, raster, lon = NULL, lat = NULL)
```

Arguments

`size`	Number or fraction per raster class.
`raster`	A categorical `SpatRaster`.
`lon, lat`	Optional coordinate column names.

Value

A `gedi_sampling_method` object containing the raster-stratified sampling parameters for use with `sample()`.

readGEDIArchiveRegion *Read the footprint boundary and forest mask for a regional archive*

Description

Read the footprint boundary and forest mask for a regional archive

Usage

```
readGEDIArchiveRegion(root)
```

Arguments

root	Archive directory with the optional Amazon region assets.

Value

A regional filter, or NULL for an archive without region assets.

readLevel1B *Read GEDI Level1B data (Geolocated Waveforms)*

Description

This function reads GEDI level1B products: geolocated Waveforms

Usage

```
readLevel1B(level1Bpath)
```

Arguments

level1Bpath	One or more local paths or Earthdata Cloud URLs pointing to GEDI Level 1B HDF5 granules.

Value

A `gedi.level1b` object for one granule, or a lazy multi-granule collection consumed by `getLevel1BGeo()` and `getLevel1BWF()`.

See Also

`hdf5r::H5File` in the *hdf5r* package and https://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi01-b-002

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`	One or more local paths or Earthdata Cloud URLs pointing to GEDI Level 2A HDF5 granules, or a gediDB reference returned by `gedifinder()` with `source = "gedidb"`.

Value

An S4 object of class `gedi.level2a` for HDF5 input, or a lazy `gedidb_level2a` object for a gediDB reference. Multiple paths return a lazy collection consumed by `getLevel2AM()`.

See Also

https://www.earthdata.nasa.gov/data/catalog/lpcloud-gedi02-a-002

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`	One or more local paths or Earthdata Cloud URLs pointing to GEDI Level 2B HDF5 granules, or a gediDB reference returned by `gedifinder()`.

Value

A `gedi.level2b` object for one HDF5 granule, a lazy remote object, or a multi-granule collection consumed by `getLevel2BVPM()`.

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)
```

readLevel3 *Read GEDI Level 3 gridded metrics*

Description

Read GEDI Level 3 gridded metrics

Usage

```
readLevel3(x, metrics = NULL)
```

Arguments

x	One or more local paths, GDAL-readable URLs, or a SpatRaster.
metrics	Optional layer names to retain.

Value

A terra::SpatRaster.

readLevel4A *Read a GEDI Level 4A granule*

Description

Opens a GEDI04_A HDF5 granule containing footprint-level aboveground biomass density estimates.

Usage

```
readLevel4A(level4Apath)
```

Arguments

level4Apath	One or more local paths or Earthdata Cloud URLs pointing to GEDI04_A HDF5 granules, or a gediDB reference returned by gedifinder().

Value

A gedi.level4a object for HDF5 input, or a lazy gedidb_level4a object for a gediDB reference. Multiple paths return a lazy collection consumed by getLevel4A(). Close HDF5 inputs with close() and cached gediDB providers with gedidb_disconnect().

readLevel4B *Read GEDI Level 4B gridded biomass*

Description

Read GEDI Level 4B gridded biomass

Usage

```
readLevel4B(x, metrics = NULL)
```

Arguments

x	One or more local paths, GDAL-readable URLs, or a SpatRaster.
metrics	Optional layer names to retain.

Value

A terra::SpatRaster for local or GDAL-readable data, or a lazy gedi_gee_object for a Google Earth Engine reference.

See Also

doi:10.3334/ORNLDAAC/2299

resampleLevel3 *Resample a GEDI Level 3 raster*

Description

Resample a GEDI Level 3 raster

Usage

```
resampleLevel3(x, template, method = "bilinear", ...)
```

Arguments

x	A raster path or SpatRaster.
template	Target SpatRaster.
method	Resampling method.
...	Passed to terra::resample().

Value

A terra::SpatRaster.

resampleLevel4B *Resample a GEDI Level 4B raster*

Description

Resample a GEDI Level 4B raster

Usage

```
resampleLevel4B(x, template, method = "bilinear", ...)
```

Arguments

x	A raster path or SpatRaster.
template	Target SpatRaster.
method	Resampling method.
...	Passed to terra::resample().

Value

A terra::SpatRaster.

rGEDI_configure *Configure optional rGEDI cloud and Earth Engine support*

Description

Configure optional rGEDI cloud and Earth Engine support

Usage

```
rGEDI_configure(
  python = NULL,
  install = FALSE,
  gedidb = FALSE,
  earth_engine_project = Sys.getenv("EE_PROJECT", unset = ""),
  quiet = FALSE
)
```

Arguments

python	Optional Python executable or environment name.
install	Logical; install missing Python modules.
gedidb	Logical; include the optional gediDB query backend.
earth_engine_project	Optional Google Earth Engine project ID.
quiet	Logical.

Value

Invisibly returns a capability table.

`sample` *Sample GEDI footprints or ordinary R objects*

Description

A GEDI footprint table paired with a GEDI sampling specification uses the spatial sampling implementation. All other calls are forwarded unchanged to `base::sample()`.

Usage

```
sample(x, ...)
```

Arguments

`x`	A GEDI footprint table or an object accepted by `base::sample()`.
`...`	A `gedi_sampling_method` object for GEDI tables, or the standard `size`, `replace`, and `prob` arguments for `base::sample()`.

Value

A data.table::data.table for GEDI sampling; otherwise the value returned by `base::sample()`.

`search_datasets` *Search the GEDI datasets available in Earth Engine*

Description

Search the GEDI datasets available in Earth Engine

Usage

```
search_datasets(..., operator = c("and", "or"))
```

Arguments

`...`	Words matched against the product name, catalog ID, and title.
`operator`	Use `"and"` or `"or"` when several words are supplied.

Value

A data table containing product, catalog ID, version, and title.

slope *Earth Engine terrain slope*

Description

Earth Engine terrain slope

Usage

```
slope(x)
```

Arguments

x	Earth Engine elevation image.

Value

An Earth Engine image.

spacedSampling *GEDI minimum-distance sampling specification*

Description

GEDI minimum-distance sampling specification

Usage

```
spacedSampling(size, radius, lon = NULL, lat = NULL)
```

Arguments

size	Number or fraction of observations to select.
radius	Minimum separation in meters.
lon, lat	Optional coordinate column names.

Value

A `gedi_sampling_method` object containing the minimum-distance sampling parameters for use with `sample()`.

stratifiedSampling *GEDI attribute-stratified sampling specification*

Description

GEDI attribute-stratified sampling specification

Usage

```
stratifiedSampling(size, variable, breaks = NULL)
```

Arguments

size	Number or fraction per stratum.
variable	Column used as strata, or a numeric column to bin.
breaks	Optional breaks for a numeric variable.

Value

A gedi_sampling_method object containing the attribute-stratified sampling parameters for use with sample().

to_vect *Convert GEDI footprints to a spatial vector*

Description

Convert GEDI footprints to a spatial vector

Usage

```
to_vect(x, lon = NULL, lat = NULL, crs = "EPSG:4326")
```

Arguments

x	GEDI footprint table.
lon, lat	Optional coordinate column names.
crs	Coordinate reference system.

Value

A terra::SpatVector.

varSel *Select predictor variables for a GEDI model*

Description

Select predictor variables for a GEDI model

Usage

```
varSel(
  x,
  y,
  method = c("correlation", "randomForest", "rfe"),
  threshold = 0.1,
  seed = NULL,
  ...
)
```

Arguments

x	Predictor data frame.
y	Response vector.
method	Selection method. "rfe" recursively removes the least important predictor and selects the smallest model whose out-of-bag error is within one standard error of the minimum.
threshold	Minimum absolute correlation or scaled Random Forest importance. For method = "rfe", it is applied after model selection; use 0 to retain the RFE-selected subset.
seed	Optional random seed.
...	Passed to randomForest when requested.

Value

An object of class gedi_var_selection.

vect_as_ee *Convert a vector object to an Earth Engine feature collection*

Description

Convert a vector object to an Earth Engine feature collection

Usage

```
vect_as_ee(x)
```

Arguments

x	sf, sfc, SpatVector, or vector file path.

Value

An Earth Engine FeatureCollection.

Index

accumulateGEDIGranule, 5
accumulateGEDIGranule(), *6*
accumulateGEDIGranules, 5
addEEImage, 6
aggregateLevel3, 7
aggregateLevel4B, 7
aspect, 8
assembleGEDIArchiveYears, 8

base::sample(), *101*
build_ee_forest, 9

clip, 9
clip,data.frame,ANY-method *(*clip*)*, 9
clip,data.frame,numeric-method *(*clip*)*, 9
clip,data.frame,SpatExtent-method *(*clip*)*, 9
clip,gedi.level1b,ANY-method *(*clip*)*, 9
clip,gedi.level1b,numeric-method *(*clip*)*, 9
clip,gedi.level1b,SpatExtent-method *(*clip*)*, 9
clip,gedi.level2a,ANY-method *(*clip*)*, 9
clip,gedi.level2a,numeric-method *(*clip*)*, 9
clip,gedi.level2a,SpatExtent-method *(*clip*)*, 9
clip,gedi.level2b,ANY-method *(*clip*)*, 9
clip,gedi.level2b,numeric-method *(*clip*)*, 9
clip,gedi.level2b,SpatExtent-method *(*clip*)*, 9
clip,gedi.level4a,ANY-method *(*clip*)*, 9
clip,gedi.level4a,numeric-method *(*clip*)*, 9
clip,gedi.level4a,SpatExtent-method *(*clip*)*, 9
clip,gedi_gee_object,ANY-method *(*clip*)*, 9
clip,SpatRaster,ANY-method *(*clip*)*, 9
clipLevel1B, 11
clipLevel1BGeo, 13
clipLevel1BGeoGeometry, 14
clipLevel1BGeometry, 16
clipLevel2A, 17
clipLevel2AGeometry, 18
clipLevel2AM, 20
clipLevel2AMGeometry, 21
clipLevel2B, 22
clipLevel2BGeometry, 24
clipLevel2BPAIProfile, 25
clipLevel2BPAIProfileGeometry, 27
clipLevel2BPAVDProfile, 28
clipLevel2BPAVDProfileGeometry, 29
clipLevel2BVPM, 31
clipLevel2BVPMGeometry, 32
clipLevel3, 34
clipLevel4A, 34
clipLevel4AGeometry, 35
clipLevel4B, 36
close(), *37*, *98*
close,gedi.level1b-method, 36
close,gedi.level2a-method *(*close,gedi.level1b-method*)*, 36
close,gedi.level2b-method *(*close,gedi.level1b-method*)*, 36
close,gedi.level4a-method *(*close,gedi.level1b-method*)*, 36
close.gedi.cloud_h5, 37

data.table::data.table, *11*, *13–15*, *20*, *21*, *26–28*, *30*, *31*, *33*, *35*, *42*, *50*, *51*, *57*, *60*, *62*, *63*, *65*, *66*, *68*, *70*, *87–89*, *101*

earthdata_login, 37
earthdata_s3_credentials, 38
ee_build_AlphaEarth_embedding_terrain_stack, 38
ee_build_hls_s1c_terrain_stack, 39
ee_check_task_status, 39

ee_check_task_status(), *41*
ee_image_to_drive, 40
ee_initialize, 41
ee_rect_to_sf, 41
ext_to_ee, 42
ext_to_ee(), *40*
extractEE, 42
extractEE(), *66*, *68*
extractGEDIAncillary *(*extractEE*)*, 42

fit_metrics, 43
fit_model, 43
fit_model(), *91*

gedi.fullwaveform, *78*
gedi.fullwaveform
*(*gedi.fullwaveform-class*)*, 44
gedi.fullwaveform-class, 44
gedi.level1b, *12*, *16*, *36*, *51*, *53–55*, *95*
gedi.level1b *(*gedi.level1b-class*)*, 44
gedi.level1b-class, 44
gedi.level2a, *96*
gedi.level2a *(*gedi.level2a-class*)*, 45
gedi.level2a-class, 45
gedi.level2b, *97*
gedi.level2b *(*gedi.level2b-class*)*, 45
gedi.level2b-class, 45
gedi.level4a, *35*, *67*, *98*
gedi.level4a *(*gedi.level4a-class*)*, 45
gedi.level4a-class, 45
gedidb_available, 46
gedidb_disconnect, 46
gedidb_disconnect(), *98*
gediDownload, 47
gediEE, 48
gedifinder, 49
gedifinder(), *96–98*
gediWFMetrics, 50
gediWFSimulator, 51
gee_available, 53
geomSampling, 54
get_catalog_id, 70
getLevel1BGeo, 54
getLevel1BGeo(), *13*, *14*, *55*, *95*
getLevel1BWF, 58
getLevel1BWF(), *78*, *95*
getLevel2AM, 59
getLevel2AM(), *72*, *88*, *96*
getLevel2BPAIProfile, 61
getLevel2BPAIProfile(), *26*, *27*, *29*, *81*, *82*
getLevel2BPAVDProfile, 62
getLevel2BPAVDProfile(), *28*, *83*
getLevel2BVPM, 64
getLevel2BVPM(), *73*, *89*, *97*
getLevel3, 66
getLevel4A, 67
getLevel4A(), *35*, *75*, *80*, *90*, *98*
getLevel4B, 68
getTileUrl, 69
getTrack, 69
ggplot2::ggplot, *82*, *83*
glcmTexture, 71
graphics::clip(), *11*
graphics::plot(), *43*, *80*
gridSampling, 71
gridStatsLevel2AM, 72
gridStatsLevel2BVPM, 73
gridStatsLevel4A, 75

H5File, *44*, *45*
hdf5r::H5File, *16*, *95*

leaflet::addTiles(), *6*

map_create, 75
map_download, 76
map_view, 76
mosaicLevel3, 77
mosaicLevel4B, 77

plot,gedi.fullwaveform,missing-method,
78
plot.gedi_var_selection, 79
plot_gedi_orbit_animation, 85
plot_gedi_orbit_animation(), *69*
plotLevel3, 80
plotLevel4A, 80
plotLevel4B, 81
plotPAIProfile, 81
plotPAVDProfile, 83
plotWFMetrics, 84
polyStats, 87
polyStatsLevel2AM, 87
polyStatsLevel2BVPM, 89
polyStatsLevel4A, 90
predict.gedi_model, 91
predictGEDIH5, 91
projectLevel3, 92

projectLevel4B, 93

randomSampling, 93
rasterize, 94
rasterSampling, 94
readGEDIArchiveRegion, 95
readGEDIArchiveRegion(), *5, 6*
readLevel1B, *58*, 95
readLevel1B(), *12*, *16*, *31, 32*, *58*, *84*
readLevel2A, 96
readLevel2A(), *17*, *19–21*, *60*, *84*
readLevel2B, 97
readLevel2B(), *23, 24*, *61*, *63, 64*
readLevel3, 98
readLevel3(), *66*
readLevel4A, 98
readLevel4A(), *35*, *67*
readLevel4B, 99
readLevel4B(), *68*
resampleLevel3, 99
resampleLevel4B, 100
rGEDI-package, 4
rGEDI_configure, 100

sample, 101
sample(), *54*, *71*, *93, 94*, *102, 103*
search_datasets, 101
sf::sf, *14*, *19*, *21*, *24*, *27*, *30*, *33*
sf::st_read, *14*, *21*, *24*, *27*, *30*, *33*
sf::st_read(), *19*
slope, 102
spacedSampling, 102
stats::predict(), *91*
stratifiedSampling, 103

tempdir(), *47*
terra::aggregate(), *7*
terra::as.data.frame(), *66*, *68*
terra::extract(), *66*, *68*, *87*
terra::plot(), *80, 81*
terra::project(), *92, 93*
terra::rasterize(), *94*
terra::resample(), *99, 100*
terra::SpatExtent, *10*
terra::SpatRaster, *7*, *10, 11*, *34*, *36*, *75*, *77, 78*, *87*, *92–94*, *98–100*
terra::SpatVector, *11*, *103*
terra::vect(), *66*, *68*, *87*
to_vect, 103

varSel, 104
vect_as_ee, 104