Package: lidarHD (via r-universe)

November 12, 2026

Type Package

Version 1.2.1

Title Utilities to Download and Manage Files of France National Airborne Laser Scanning Cover (LIDAR HD)

Date 2026-11-03

Description Provides functions for downloading, managing and processing (normalization, digital elevation models computation) laz files provided by the LiDAR HD program of IGN (France).

URL https://lidar.pages-forge.inrae.fr/lidarHD/, https://forge.inrae.fr/lidar/lidarHD

BugReports https://forge.inrae.fr/lidar/lidarHD/-/issues

Depends R (>= 4.0.0)

Imports methods, utils, archive, sf, terra, future, future.apply, lidR (>= 4.2.3), rlas (>= 1.8.4), rstac

Suggests lasR

License GPL-3

LazyData TRUE

Encoding UTF-8

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libabsl-dev cmake libfreetype6-dev libgdal-dev gdal-bin libgeos-dev libglu1-mesa-dev make texlive libarchive-dev libjpeg-dev libpng-dev libuv1-dev libgl1-mesa-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

Repository https://fabricevinatierpro-hub.r-universe.dev

Date/Publication 2026-11-03 11:05:31 UTC

RemoteUrl https://forge.inrae.fr/lidar/lidarHD.git

RemoteRef HEAD

RemoteSha b97daca7395d12551f89d09a3023c238bd454748

Contents

add_attributes_ta . . . 2
add_height_catalog . . . 3
add_height_extrabyte . . . 4
catalog_bloc . . . 4
catalog_folder . . . 5
check_folder . . . 5
check_folder_string . . . 6
compress_tif . . . 7
compute_dtm . . . 7
create_lax . . . 8
download_files . . . 9
download_files_ch . . . 10
download_mnx . . . 11
eb2nom . . . 12
flag_processed . . . 13
folder2bloc . . . 13
generer_dates_valides . . . 14
laz_name2tif_name . . . 15
load_classified_ta . . . 15
load_classified_ta_ch . . . 16
load_mnx . . . 17
load_saved_ta . . . 17
normalize_files . . . 18
OVH2GP . . . 19
process_files . . . 19
url2annee . . . 20
url2bloc . . . 21
url2meta . . . 21

Index **22**

`add_attributes_ta` *Ajouter les attributs des produits derives au tableau d'assemblage*

Description

Fonction qui ajoute dans de tableau d'assemblage le chemin d'acces aux produits derives

Usage

```
add_attributes_ta(ta, data_folder, save = FALSE)
```

Arguments

`ta`	text or 'sf' object. 'sf' object or filename to load it
`data_folder`	test. path to directory with downloaded and computed files
`save`	boolean. in case 'ta' is a file, should it be overwritten ?

Value

An ‘sf‘ object with additional attributes containing the file path to derived products

Examples

```
## Not run:
# charger depuis un fichier presente dans un dossier
ta_lidarHD <- load_classified_ta(url = NULL, save_dir = "./data/sig/")
ta_lidarHD <- lidarHD::add_attributes_ta(ta_lidarHD, dossier, save = FALSE)

## End(Not run)
```

---

add_height_catalog *Add Height Above Ground to a selection of files in a catalog*

Description

Adds the height above ground in an ExtraByte attribute `H` to files of a catalog. Files are renamed with the prefix "EB-" and are indexed with a ".lax" file because they are not written as "copc" files. Applies the function `add_height_extrabyte`, and requires that adjacent tiles are present in the catalog to avoid border effects.

Usage

```
add_height_catalog(cata, roi, prompt = TRUE, ncores = 2)
```

Arguments

`cata`	A `LAScatalog-class` object. In case no region of interest is provided and the catalog object has a "processed" attribute, only files with `TRUE` values are processed.
`roi`	`sf` object or character vector. Region of interest if `sf` object, "blocs" of interest if character vector.
`prompt`	boolean. Should the user be prompted before plotting and normalizing ?
`ncores`	Integer, number of cores to use with `future` package.

Value

The updated `LAScatalog-class` object.

Examples

```
## Not run: add_height_catalog
```

add_height_extrabyte *Add Height above ground as extrabyte attribute*

Description

Adds the height above ground in an ExtraByte attribute H to objects of class LAS-class or to files recorded in an object of class LAScatalog-class, by applying the function normalize_height

Usage

```
add_height_extrabyte(las)
```

Arguments

las	An object of class LAS-class or LAScatalog-class

Value

A LAS-class object with the H attribute in the case. In case of a LAScatalog-class input, the function is used for the side effect of writing files as specified in the catalog options.

Examples

```
## Not run: add_height_extrabyte
```

catalog_bloc *Creer et sauvegarder le catalogue d'un bloc*

Description

Creer un objet cata au format LAScatalog-class avec les fichiers LAZ situes dans un dossier et le sauvegarde dans le fichier cata.rda a l'interieur du dossier

Usage

```
catalog_bloc(folder, bloc, save_on_disk = TRUE)
```

Arguments

folder	text. folder to analyse
bloc	text. bloc to catalog
save_on_disk	boolean. Save catalog on disk or not

Value

A LAS catalog object

Examples

```
## Not run: catalog_bloc("./LidarHD/data/LAZ_classe/", "OE")
```

`catalog_folder` *Creer et sauvegarder le catalogue du dossier complet*

Description

Charge les objets au format `LAScatalog-class` presents dans les sous-dossiers (fichiers `cata.rda`) et les fusionne dans un objet `cata` au format `LAScatalog-class` sauvegarde dans le fichier `cata.rda` a la racine du dossier. Si `blocs` est specifie, le catalogue est mis a jour uniquement pour les fichiers de ces blocs

Usage

```
catalog_folder(folder, blocs = NULL, save_on_disk = TRUE)
```

Arguments

`folder`	character. folder to analyse
`blocs`	character vector. subset of blocs to update
`save_on_disk`	boolean. Save catalog on disk or not

Value

A LAS catalog object

Examples

```
## Not run: catalog_folder("./LidarHD/data/LAZ_classe/")
```

`check_folder` *Verifier les fichiers telecharges*

Description

Verifie que les fichiers laz sont dans le dossier correspondant a leur bloc. Verifie que les catalogues des dossiers (fichiers `cata.rda` a la racine, contenant l'objet `cata` au format `LAScatalog-class`) correspondent aux fichiers presents.

Usage

```
check_folder(folder, ta = NULL)
```

Arguments

folder	text. Folder to analyse
ta	sf object. Output of load_classified_ta

Value

No return, the function is used to check folder

See Also

catalog_bloc to force updating of LAScatalog of a bloc, catalog_folder to force updating LAScatalog of all blocs from the saved LAScatalogs in each bloc, create_lax to generate missing index files.

Examples

```
## Not run: check_folder("./LidarHD/data/LAZ_classe/")
```

---

check_folder_string *Verifie que le nom de dossier finit par /*

Description

Ajoute un "/" a un chemin de dossier si ce n'est pas le cas

Usage

```
check_folder_string(folder)
```

Arguments

folder	text. folder path

Value

A string

Examples

```
check_folder_string("./LidarHD/data/LAZ_classe/LN")
```

`compress_tif` *Compresser un fichier tif*

Description

Compresser un fichier tif (options COMPRESS=LZW, PREDICTOR=3)

Usage

```
compress_tif(file)
```

Arguments

`file`	text. filename

Value

A boolean

`compute_dtm` *Calcul de MNT a partir des fichiers LiDAR HD*

Description

Calcul des fichiers MNT a partir des fichiers LiDAR HD de l'IGN presents dans un dossier (par exemple telecharges avec `download_files`) et les enregistre dans un autre repertoire. Les fichiers MNT deja presents ne sont pas re-calcules. Le calcul est effectuee avec la fonction `rasterize_terrain` et l'algorithme `tin` avec les points de classe 2 et 9. Les fichiers a traiter peuvent etre selectionnes via une zone d'interet, il est conseille que les fichiers adjacents soient egalement telecharges pour eviter les effets de bord.

Usage

```
compute_dtm(
  laz_folder,
  dtm_folder = NULL,
  res = 1,
  roi = NULL,
  prompt = TRUE,
  ncores = 2L
)
```

Arguments

laz_folder	Text, path to the folder that contains the laz files downloaded from IGN. The folder should contain a "cata.rda" file with the saved LAScatalog-class object of files in the folder.
dtm_folder	Text, path to the folder that contains or will contain the DTM files. If NULL a folder is created in the same directory as ‘laz_folder‘ with name "DTM_#res#m" where #res# is parameter ‘res‘.
res	numeric, DTM resolution
roi	sf object or character vector. Region of interest if sf object, "blocs" of interest if character vector.
prompt	boolean. Should the user be prompted before plotting and computing ?
ncores	Integer, number of cores to use with future package.

Value

A data.frame with the files to compute the DTM from and a boolean indicating if the computation was successful or not.

Examples

```
## Not run: compute_dtm("./data/LAZ_classe/", res = 1, prompt = TRUE, ncores = 2L)
```

---

create_lax *Indexer les fichiers d'un repertoire*

Description

Verifie si les fichiers laz dans un dossier ont leur index lax. Propose d'indexer les fichiers qui ne le sont pas. Les fichiers au format copc (par exemple les nuages LiDAR HD fournis par l'IGN) ne sont pas indexes.

Usage

```
create_lax(folder, lasindex_path = NULL, proceed = FALSE, ncores = 4L)
```

Arguments

folder	text. folder to analyse
lasindex_path	text. if set to NULL, the function write.las is used to produce the index. Indicate the path to lasindex bash program from LAStools if you want to use it instead (tested only on Linux).
proceed	boolean. Automatically index files (TRUE) or ask user
ncores	integer. Number of cores to use in a multisession of plan

Value

Returns the number of indexed files

Examples

```
## Not run: create_lax("./LidarHD/data/LAZ_classe/OE/")
```

`download_files` *Telecharger les dalles lidarHD dans une zone d'interet*

Description

Selectionne les dalles qui intersectent une region d'interet, ou celles dans les blocs specifies. Propose de les telecharger si elles ne sont pas deja dans un repertoire indique par l'utilisateur, un sous-dossier est cree par bloc. Les fichiers sont deja indexes (format copc). Un objet `cata` au format `LAScatalog-class` est sauvegarde dans chaque sous-dossier bloc, fichier `cata.rda`. Un objet `cata` pour l'ensemble des blocs est sauvegarde a la racine du dossier.

Usage

```
download_files(ta, roi = NULL, folder = NULL, buffer = TRUE, prompt = TRUE)
```

Arguments

`ta`	`sf` object. Tiles available from IGN (output of `load_classified_ta`
`roi`	`sf` object or character vector. Region of interest if `sf` object, "blocs" of interest if character vector
`folder`	text. Local folder for laz files. Already downloaded files should not have their names changed. They are placed in sub-directories corresponding to blocs.
`buffer`	boolean. Should tiles adjacent to tiles intersecting the roi be downloaded to avoid edge artefacts in further processing ?
`prompt`	boolean. Should the user be prompted before plotting and downloading ?

Value

The input object with an additional attribute containing a text string indicated if tiles are in (dedans) or out (dehors) of roi, and if they are already in the target `folder` done (fait) or todo (a_faire)). This information is not updated in case the user asks for tiles to be downloaded.

See Also

`load_classified_ta` to download available tiles, `check_folder` to check that the structure of the folder which contains downloaded files is correct

Examples

```
## Not run:
ta <- load_classified_ta()
download_files(ta, folder = "./LidarHD/data/LAZ_classe/")

## End(Not run)
```

`download_files_ch` *Telecharger les dalles swisstopo dans une zone d'interet (CH) A TERMINER*

Description

Verifie les dalles qui intersectent une region d'interet Propose de les telecharger si elles ne sont pas deja dans un repertoire indique par l'utilisateur

Usage

```
download_files_ch(
  ta,
  roi = NULL,
  las2las_path = "~/LAStools/bin/las2las",
  year = NULL,
  folder = NULL,
  buffer = TRUE
)
```

Arguments

`ta`	sf object. Tiles available from swisstopo
`roi`	sf object. Region of interest
`las2las_path`	text. path to the `las2las` bash program from LAStools
`year`	numeric. Year of data
`folder`	text. Local folder for laz files. Already downloaded files should not have their names changed
`buffer`	boolean. Should tiles adjacent to tiles intersecting the roi be downloaded to avoid edge artefacts in further processing ?

Value

The input object with an additional attribute containing "out_of_roi", "in_todo", "in_done". This information is not updated in case the user asks for "in_todo" tiles to be downloaded

download_mnx *Telecharger les MNX dans une zone d'interet*

Description

Selectionne les dalles MNX qui intersectent une region d'interet. Propose de les telecharger si elles ne sont pas deja dans un repertoire indique par l'utilisateur

Usage

```
download_mnx(
  ta,
  mnx = "mnt",
  roi = NULL,
  folder = NULL,
  buffer = TRUE,
  prompt = TRUE,
  compress = TRUE,
  workers = 2
)
```

Arguments

ta	sf object. Tiles available from IGN (output of load_classified_ta
mnx	text. which mnx to download (mnh, mns or mnt)
roi	sf object. Region of interest
folder	text. Local folder for files. Already downloaded files should not have their names changed.
buffer	boolean. Should tiles adjacent to tiles intersecting the roi be downloaded to avoid edge artefacts in further processing ?
prompt	boolean. Should the user be prompted before plotting and downloading ?
compress	boolean. Should files be compressed (LZW with predictor = 3) ?
workers	integer. Number of cores to use in future::plan(future::multisession(workers =))

Value

The input object with an additional attribute containing a text string indicating the link to the local file.

See Also

load_classified_ta to download available tiles,

Examples

```
## Not run:
# definir le roi
roi <- c(xmin = 5.72, xmax = 5.75, ymin =  45.11, ymax = 45.14)
roi <- sf::st_bbox(roi, crs = 4326)
roi <- sf::st_as_sfc(roi)
roi <- sf::st_transform(roi, 2154)
# charger le tableau
ta <- load_classified_ta(url = "https://data.geopf.fr/wfs/ows?SERVICE=WFS&REQUEST=GetFeature&VERSION=2.0.0&TYPEN
# telecharger les mnt
download_mnx(ta, "mnt", roi = roi, folder = "./LidarHD/data/MNT/")

## End(Not run)
```

---

eb2nom *Convertir le nom EB en nom de la dalle d'origine*

Description

Fonction qui reconstruit le nom de la dalle du nom du fichier ExtraByte

Usage

```
eb2nom(nom_eb)
```

Arguments

nom_eb	text. Tile download url

Value

A string

Examples

```
eb2nom("./LidarHD/data/LAZ_classe/LN/EB-LHD_FXX_0686_6384_PTS_C_LAMB93_IGN69.laz")
```

flag_processed — *Marque comme processed les dalles d'un catalogue dans zone d'interet*

Description

Ajoute un attribut processed dans un objet au format LAScatalog-class pour les fichiers qui intersectent une zone d'interet definie par le nom des blocs ou une emprise spatiale.

Usage

```
flag_processed(cata, roi = NULL, keep_all = FALSE)
```

Arguments

cata	LAScatalog-class object.
roi	sf object or character vector. Region of interest if sf object, "blocs" of interest if character vector
keep_all	boolean. Should all tiles be kept in LAScatalog, or only tiles relevant for processing ?

Value

LAScatalog-class object with additionnal processed attribute.

See Also

download_files to download tiles,

Examples

```
## Not run:
flag_processed(cata, "OE")

## End(Not run)
```

folder2bloc — *Extraire le bloc du nom de dossier*

Description

Fonction qui extrait le nom du bloc a partir du dossier de stockage

Usage

```
folder2bloc(folder)
```

Arguments

folder	text. folder names

Value

A string

Examples

```
## Not run: url2meta("https://data.geopf.fr/telechargement/download/LiDARHD-NUALID/NUALHD_1-0__LAZ_LAMB93_PM_20
```

---

generer_dates_valides *Generer des dates valides au format AAAA-MM-JJ*

Description

Cette fonction cree toutes les combinaisons possibles d'annees, de mois et de jours et retourne uniquement les dates valides du calendrier (en excluant les jours inexistants comme le 31 avril ou le 29 fevrier d'une annee non bissextile).

Usage

```
generer_dates_valides(years, months, days)
```

Arguments

years	Un vecteur numerique d'annees (ex: 'c(2023, 2024)').
months	Un vecteur numerique de mois (1 a 12).
days	Un vecteur numerique de jours (1 a 31).

Value

Un vecteur de chaines de caracteres representant les dates valides au format '"AAAA-MM-JJ"'.

Examples

```
# Exemple de base
generer_dates_valides(years = c(2023, 2024), months = 1:3, days = 28:31)

# Limiter a une seule annee
generer_dates_valides(years = 2025, months = 2, days = 27:29)
```

laz_name2tif_name *Convertir le nom de dalle laz en nom de fichier MNT*

Description

Fonction qui convertit un nom de dalle en nom de fichier MNT

Usage

```
laz_name2tif_name(nom, res, dtm_folder)
```

Arguments

nom	text. Laz file full filename
res	numeric. resolution for dtm file
dtm_folder	test. Target folder for files

Value

A string

Examples

```
## Not run: create_lax("/media/reseau/drones/Lidar/LidarHD/data/LAZ_classe/")
```

load_classified_ta *Charger le tableau d'assemblage des metadonnees LiDARHD IGN*

Description

Si une url WFS est fournie : telecharge le tableau d'assemblage des dalles : catalogue stac Theia par defaut (https://api.stac.teledetection.fr), peut telecharger egalement le tableau d'assemblage des metadonnes IGN Sinon charge un gpkg sauvegarde precedemment dans le dossier

Usage

```
load_classified_ta(
  url = "https://api.stac.teledetection.fr",
  save_dir = NULL,
  bbox = NULL
)
```

Arguments

url	text. link of the wfs source, if set to NULL data will be loaded from the most recent file in save_dir with load_saved_ta.
save_dir	text. if specified, directory where the output is saved, with name ta_YYMMDD.gpkg
bbox	sf object from which a bbox can be extracted, with crs information

Value

An sf object of the tiles

Examples

```
## Not run:
# from catalog stac
# bbox exemple
bbox <- c(xmin = 5.72, xmax = 5.75, ymin =  45.11, ymax = 45.14)
# bbox France metrolpolitaine
bbox <- c(xmin = -5.14, ymin = 41.35, xmax = 9.56, ymax = 51.10)
# ajouter le crs
bbox <- sf::st_bbox(bbox, crs = sf::st_crs(4326))
ta <- load_classified_ta(bbox = bbox)
# from WFS IGN
ta <- load_classified_ta(url="https://data.geopf.fr/wfs/ows?SERVICE=WFS&REQUEST=GetFeature&VERSION=2.0.0&TYPENA
## End(Not run)
```

---

load_classified_ta_ch *Charger le tableau d'assemblage des nuages classifies swisstopo (CH)*

Description

Telecharger manuellement la liste des fichiers sur https://www.swisstopo.admin.ch/fr/geodata/height/surface3d.html selection sur la carte avec un polygone qui englobe la suisse decocher "etat temporel" cocher "publie avant" et mettre la date du jour

Usage

```
load_classified_ta_ch(
  file =
   "/media/reseau/drones/Lidar/Swisstopo/data/SIG/ch.swisstopo.swisssurface3d-wNDKT0QW.csv"
)
```

Arguments

file	text. file downloaded from swisstopo

Value

A sf object of the classified tiles

Examples

```
## Not run: ta <- load_classified_ta_ch()
```

`load_mnx` *Charger un MNT sur une emprise*

Description

Charge le MNT a 0.5 m de resolution sur l'emprise specifiee a partir du flux wms-r Ajuste l emprise sur le maillage MNT (origine a 0.25, 0.25)

Usage

```
load_mnx(emprise, mnx = "MNT")
```

Arguments

`emprise`	vector with xmin xmax ymin max
`mnx`	string type of mnx in c("MNT", "MNS", "MNH")

Value

a raster.

See Also

`load_classified_ta` and `download_mnx` to download available tiles,

Examples

```
mnh <- load_mnx(c(917400, 917500, 6469400, 6469500), "MNH")
mnh
```

`load_saved_ta` *Charger le tableau d'assemblage sauvegarde*

Description

Charge le fichier sauvegarde le plus recent dans un repertoire

Usage

```
load_saved_ta(folder = NULL)
```

Arguments

folder	string. Folder containing the saved files, with pattern ta_YYMMDD.gpkg

Value

An sf object of the classified tiles and blocs

Examples

```
## Not run: ta <- load_saved_ta("./data/sig/ta_231218.gpkg")
```

---

normalize_files *Normalisation des fichiers LiDAR HD*

Description

Normalise les fichiers LiDAR HD de l'IGN presents dans un dossier (par exemple telecharges avec download_files) et les enregistre dans un autre repertoire. Les fichiers normalises deja presents ne sont pas re-calcules. La normalisation est effectuee avec la fonction normalize_height et l'algorithme tin. Les fichiers a normaliser peuvent etre selectionnes via une zone d'interet, il est conseille que les fichiers adjacents soient egalement telecharges pour eviter les effets de bord.

Usage

```
normalize_files(original, normalized, roi = NULL, prompt = TRUE, ncores = 2L)
```

Arguments

original	Text, path to the folder that contains the laz files downloaded from IGN. The folder should contain a "cata.rda" file with the saved LAScatalog-class object of files in the folder.
normalized	Text, path to the folder that contains or will contain the normalized laz files. If some files are already present, the folder should contain a "cata.rda" file with the saved LAScatalog-class object of files in the folder.
roi	sf object or character vector. Region of interest if sf object, "blocs" of interest if character vector.
prompt	Boolean. Should the user be prompted before plotting and normalizing ?
ncores	Integer, number of cores to use with future package.

Value

The updated LAScatalog-class object of normalized files in the normalized folder.

Examples

```
## Not run: normalize_files("./data/LAZ_classe/", "./data/LAZ_norm/", prompt = TRUE, ncores = 2L)
```

OVH2GP *Convertir le nom de fichier OVH en nom geopplateforme*

Description

Modification de la chaine de caractere en enlevant "O" ou "C" et en modificant IGN69 en IGN78 pour les blocs en Corse

Usage

```
OVH2GP(fichier, bloc = NULL)
```

Arguments

fichier	Vecteur de chaine de caracteres.
bloc	Vecteur de chaine de caracteres.

Value

Un vecteur de chaines de caractères

process_files *Normalisation et calcul DTM des fichiers LiDAR HD avec lasR*

Description

Normalisation et/ou calcul de MNT avec le package lasR a partir des fichiers LiDAR HD de l'IGN presents dans un dossier (par exemple telecharges avec download_files) et les enregistre dans un autre repertoire. Les fichiers normalises ou DTM deja presents ne sont pas re-calcules.

Usage

```
process_files(
  original,
  normalized = "",
  res = NULL,
  dtm_folder = NULL,
  roi = NULL,
  prompt = TRUE,
  lasR_options = list(ncores = lasR::concurrent_files(2L))
)
```

Arguments

original	Text, path to the folder that contains the laz files downloaded from IGN. The folder should contain a "cata.rda" file with the saved LAScatalog-class object of files in the folder.
normalized	Text, path to the folder that contains or will contain the normalized laz files. If empty string, normalisation will not be done. If some files are already present, the folder should contain a "cata.rda" file with the saved LAScatalog-class object of files in the folder.
res	numeric, DTM resolution, if NULL DTM computation will not be done.
dtm_folder	Text, path to the folder that contains or will contain the DTM files. If NULL a folder is created in the same directory as 'laz_folder' with name "DTM_#res#m" where #res# is parameter 'res'.
roi	sf object or character vector. Region of interest if sf object, "blocs" of interest if character vector.
prompt	Boolean. Should the user be prompted before plotting and normalizing ?
lasR_options	list. Parameters passed to with parameter of exec function of lasR.

Value

A list with the catalog of all normalised files and/or a data.frame names of computed DTMs

Examples

```
## Not run: process_files("./data/LAZ_classe/", "./data/LAZ_norm/", prompt = TRUE, ncores = 2L)
```

---

url2annee *Extraire l'annee de l'url (CH)*

Description

Fonction qui extrait le nom de l'annee a partir de l'url de telechargement

Usage

```
url2annee(url)
```

Arguments

url	text. Tile download url or filename

Value

A dataframe with the fields

url2bloc *Extraire le nom de bloc de l'url*

Description

Fonction qui extrait le nom du bloc a partir de l'url de telechargement

Usage

```
url2bloc(url)
```

Arguments

url	text. Tile download url

Value

A string

Examples

```
## Not run: url2bloc("https://data.geopf.fr/telechargement/download/LiDARHD-NUALID/NUALHD_1-0__LAZ_LAMB93_PM_20
```

url2meta *Extraire les metadonnes de l'url (bloc, date)*

Description

Fonction qui extrait les metadonnees a partir de l'url de telechargement

Usage

```
url2meta(url)
```

Arguments

url	text. Tile download url

Value

A data.frame

Examples

```
## Not run: url2meta("https://data.geopf.fr/telechargement/download/LiDARHD-NUALID/NUALHD_1-0__LAZ_LAMB93_PM_20
```

Index

add_attributes_ta, 2
add_height_catalog, 3
add_height_extrabyte, *3*, 4

catalog_bloc, 4, *6*
catalog_folder, 5, *6*
check_folder, 5, *9*
check_folder_string, 6
compress_tif, 7
compute_dtm, 7
create_lax, *6*, 8

download_files, *7*, 9, *13*, *18*, *19*
download_files_ch, 10
download_mnx, 11, *17*

eb2nom, 12

flag_processed, 13
folder2bloc, 13

generer_dates_valides, 14

laz_name2tif_name, 15
load_classified_ta, *6*, *9*, *11*, 15, *17*
load_classified_ta_ch, 16
load_mnx, 17
load_saved_ta, *16*, 17

normalize_files, 18
normalize_height, *4*, *18*

OVH2GP, 19

plan, *8*
process_files, 19

rasterize_terrain, *7*

tin, *7*, *18*

url2annee, 20
url2bloc, 21
url2meta, 21

write.las, *8*