Package 'neonUtilities'

March 10, 2023

Version 2.2.1

Date 2023-03-08

Title Utilities for Working with NEON Data

Description NEON data packages can be accessed through the NEON Data Portal <https://www.neonscience.org> or through the NEON Data API (see <https://data.neonscience.org/data-api> for documentation). Data delivered from the Data Portal are provided as monthly zip files packaged within a parent zip file, while individual files can be accessed from the API. This package provides tools that aid in discovering, downloading, and reformatting data prior to use in analyses. This includes downloading data via the API, merging data tables by type, and converting formats. For more information, see the readme file at <https://github.com/NEONScience/NEON-utilities>.

Depends R (>= 3.6.0)

Imports httr, jsonlite, downloader, data.table, utils, R.utils, stats, tidyr, stringr, pbapply, parallel, curl

Suggests rhdf5, raster, sp, testthat

License AGPL-3

URL https://github.com/NEONScience/NEON-utilities

BugReports https://github.com/NEONScience/NEON-utilities/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

NeedsCompilation no

Author Claire Lunch [aut, cre, ctb], Christine Laney [aut, ctb], Nathan Mietkiewicz [aut, ctb], Eric Sokol [aut, ctb], Kaelin Cawley [aut, ctb], NEON (National Ecological Observatory Network) [aut]

Maintainer Claire Lunch <clunch@battelleecology.org>

Repository CRAN

Date/Publication 2023-03-10 08:40:02 UTC

R topics documented:

`byFileAOP` *Serially download all AOP files for a given site, year, and product*

Description

Query the API for AOP data by site, year, and product, and download all files found, preserving original folder structure. Downloads serially to avoid overload; may take a very long time.

Usage

```
byFileAOP(
  dpID,
  site,
  year,
  check.size = TRUE,
  savepath = NA,
  token = NA_character_
)
```

Arguments

`dpID`	The identifier of the NEON data product to pull, in the form DPL.PRNUM.REV, e.g. DP1.10023.001
`site`	The four-letter code of a single NEON site, e.g. 'CLBJ'.
`year`	The four-digit year to search for data. Defaults to 2017.
`check.size`	T or F, should the user approve the total file size before downloading? Defaults to T. When working in batch mode, or other non-interactive workflow, use check.size=F.
`savepath`	The file path to download to. Defaults to NA, in which case the working directory is used.
`token`	User specific API token (generated within neon.datascience user accounts)

Value

A folder in the working directory, containing all files meeting query criteria.

Author(s)

Claire Lunch <clunch@battelleecology.org> Christine Laney <claney@battelleecology.org>

References

Examples

```
## Not run:
# To download 2017 vegetation index data from San Joaquin Experimental Range:
byFileAOP(dpID="DP3.30026.001", site="SJER", year="2017")

## End(Not run)
```

---

byTileAOP *Download AOP tiles overlapping specified coordinates for a given site, year, and product*

Description

Query the API for AOP data by site, year, product, and tile location, and download all files found. Downloads serially to avoid overload; may take a very long time.

Usage

```
byTileAOP(
  dpID,
  site,
  year,
  easting,
  northing,
  buffer = 0,
  check.size = TRUE,
  savepath = NA,
  token = NA_character_
)
```

Arguments

dpID	The identifier of the NEON data product to pull, in the form DPL.PRNUM.REV, e.g. DP1.10023.001
site	The four-letter code of a single NEON site, e.g. 'CLBJ'.
year	The four-digit year to search for data. Defaults to 2017.
easting	A vector containing the easting UTM coordinates of the locations to download.
northing	A vector containing the northing UTM coordinates of the locations to download.
buffer	Size, in meters, of the buffer to be included around the coordinates when determining which tiles to download. Defaults to 0. If easting and northing coordinates are the centroids of NEON TOS plots, use buffer=20.
check.size	T or F, should the user approve the total file size before downloading? Defaults to T. When working in batch mode, or other non-interactive workflow, use check.size=F.
savepath	The file path to download to. Defaults to NA, in which case the working directory is used.
token	User specific API token (generated within neon.datascience user accounts)

Value

A folder in the working directory, containing all files meeting query criteria.

Author(s)

Claire Lunch <clunch@battelleecology.org> Christine Laney <claney@battelleecology.org>

References

`chem_bundles` *Bundled chemistry data product information*

Description

A dataset containing NEON data product codes of terrestrial chemistry data products and the "home" data products they are bundled with.

Usage

```
chem_bundles
```

Format

A data frame with 2 variables:

product Data product ID of a terrestrial chemistry product

homeProduct Data product ID of the corresponding home data product

Source

NEON data product bundles

`convByteSize` *Convert a number of bytes into megabytes or gigabytes*

Description

For any number of bytes, convert to a number of MB or GB

Usage

```
convByteSize(objSize)
```

Arguments

`objSize`	The size in bytes

Value

The size of the file in megabytes or gigabytes

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

`footRaster` *Extract eddy covariance footprint data from HDF5 format*

Description

Create a raster of flux footprint data. Specific to expanded package of eddy covariance data product: DP4.00200.001 For definition of a footprint, see Glossary of Meteorology: https://glossary.ametsoc.org/wiki/Footprint For background information about flux footprints and considerations around the time scale of footprint calculations, see Amiro 1998: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.922.4124&rep=rep1&type=p

Usage

```
footRaster(filepath)
```

Arguments

`filepath`	One of: a folder containing NEON EC H5 files, a zip file of DP4.00200.001 data downloaded from the NEON data portal, a folder of DP4.00200.001 data downloaded by the neonUtilities::zipsByProduct() function, or a single NEON EC H5 file. Filepath can only contain files for a single site. [character]

Details

Given a filepath containing H5 files of expanded package DP4.00200.001 data, extracts flux footprint data and creates a raster.

Value

A rasterStack object containing all the footprints in the input files, plus one layer (the first in the stack) containing the mean footprint.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

```
## Not run:
# To run the function on a zip file downloaded from the NEON data portal:
ftprnt <- footRaster(filepath="~/NEON_eddy-flux.zip")

## End(Not run)
```

---

getAPI *Get the data from API*

Description

Accesses the API with options to use the user-specific API token generated within neon.datascience user accounts.

Usage

```
getAPI(apiURL, token = NA_character_)
```

Arguments

apiURL	The API endpoint URL
token	User specific API token (generated within neon.datascience user accounts). Optional.

Author(s)

Nate Mietkiewicz <mietkiewicz@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

`getAPIHeaders` *Get only headers from API*

Description

Accesses the API with options to use the user-specific API token generated within neon.datascience user accounts.

Usage

```
getAPIHeaders(apiURL, token = NA_character_)
```

Arguments

`apiURL`	The API endpoint URL
`token`	User specific API token (generated within neon.datascience user accounts). Optional.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

`getAvg` *Get a list of the available averaging intervals for a data product*

Description

Most IS products are available at multiple averaging intervals; get a list of what's available for a given data product

Usage

```
getAvg(dpID, token = NA_character_)
```

Arguments

`dpID`	The identifier of the NEON data product, in the form DPL.PRNUM.REV, e.g. DP1.00006.001
`token`	User specific API token (generated within neon.datascience user accounts)

Value

A vector of the available averaging intervals, typically in minutes.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

```
# Get available averaging intervals for PAR data product
getAvg("DP1.00024.001")
```

---

getDatatable *Get NEON data table*

Description

This is a function to retrieve a data table from the NEON data portal for sites and dates provided by the enduser. NOTE that this function only works for NEON Observation System (OS) data products, and only for select tables

Usage

```
getDatatable(
  dpid = NA,
  data_table_name = NA,
  sample_location_list = NA,
  sample_location_type = "siteID",
  sample_date_min = "2012-01-01",
  sample_date_max = Sys.Date(),
  sample_date_format = "%Y-%m-%d",
  data_package_type = "basic",
  url_prefix_data = "https://data.neonscience.org/api/v0/data/",
  url_prefix_products = "https://data.neonscience.org/api/v0/products/",
  token = NA_character_
)
```

Arguments

dpid	character sting for NEON data product ID
data_table_name	character sting for name of the data table to download, e.g., 'sls_soilCoreCollection'

sample_location_list
: list of sites, domains, etc. If NA, retrieve all data for the given data table / dpid combination.

sample_location_type
: character sting for location type, such as 'siteID'. Must be one of the NEON controlled terms. If you're unsure, use 'siteID'

sample_date_min
: start date for query. Default is 1-Jan-2012, and this should capture the earliest NEON data record.

sample_date_max
: end date for query. Default is current date.

sample_date_format
: date format. Default/expected format is yyyy-mm-dd

data_package_type
: package type, either 'basic' or 'expanded'. If unsure, use 'expanded'

url_prefix_data
: data endpoint for NEON API.

url_prefix_products
: products endpoint for NEON API.

token
: User specific API token (generated within neon.datascience user accounts)

Value

data frame with selected NEON data

Author(s)

Eric R. Sokol <esokol@battelleecology.org>

References

---

getEddyLog *Get the full issue log set for the SAE bundle*

Description

Use the NEON API to get the issue log from all products in the bundle in a user-friendly format

Usage

```
getEddyLog(token = NA_character_)
```

Arguments

token
: User specific API token (generated within neon.datascience user accounts)

Value

A table of issues reported for the data product.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

`getFileUrls` *Get and store the file names, S3 URLs, file size, and download status (default = 0) in a data frame*

Description

Used to generate a data frame of available AOP files.

Usage

```
getFileUrls(m.urls, token = NA)
```

Arguments

`m.urls`	The monthly API URL for the AOP files
`token`	User specific API token (generated within neon.datascience user accounts)

Value

A dataframe comprised of file names, S3 URLs, file size, and download status (default = 0)

Author(s)

Claire Lunch <clunch@battelleecology.org> Christine Laney <claney@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

getIssueLog *Get the issue log for a specific data product*

Description

Use the NEON API to get the issue log in a user-friendly format

Usage

```
getIssueLog(dpID = NA, token = NA_character_)
```

Arguments

dpID	The data product identifier, formatted as DP#.#####.###
token	User specific API token (generated within neon.datascience user accounts)

Value

A table of issues reported for the data product.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

```
# Get documentation and availability of plant foliar properties data product
cfcIssues <- getIssueLog("DP1.10026.001")
```

getNeonDOI *Get either a list of NEON DOIs, or the DOI for a specific data product and release*

Description

Use the DataCite API to get NEON data DOIs in a user-friendly format

Usage

```
getNeonDOI(dpID = NA_character_, release = NA_character_)
```

Arguments

dpID	The data product identifier, formatted as DP#.#####.### [character]
release	Name of a specific release, e.g. RELEASE-2022 [character]

Value

A table of data product IDs and DOIs.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

```
## Not run:
# Get all NEON data DOIs
allDOIs <- getNeonDOI()

## End(Not run)
```

---

getPackage *Get NEON data package*

Description

Get a zipped file for a single data product, site, and year-month combination. Use the NEON data portal or API to determine data availability by data product, site, and year-month combinations.

Usage

```
getPackage(dpID, site_code, year_month, package = "basic", savepath = getwd())
```

Arguments

dpID	The identifier of the NEON data product to pull, in the form DPL.PRNUM.REV, e.g. DP1.10023.001
site_code	A four-letter NEON research site code, such as HEAL for Healy.
year_month	The year and month of interest, in format YYYY-MM.
package	Either 'basic' or 'expanded', indicating which data package to download. Defaults to basic.
savepath	The location to save the output files to

Value

A zipped monthly file

Author(s)

Christine Laney <claney@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

`getProductInfo` *Get NEON data product information*

Description

Use the NEON API to get data product information such as availability, science team, etc.

Usage

```
getProductInfo(dpID = "", token = NA)
```

Arguments

`dpID`	The data product id (optional), formated as DP#.#####.###
`token`	User specific API token (generated within neon.datascience user accounts)

Value

A named list of metadata and availability information for a single data product. If the dpID argument is omitted, a table of information for all data products in the NEON catalog.

Author(s)

Christine Laney <claney@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

```
# Get documentation and availability of plant foliar properties data product
cfcInfo <- getProductInfo("DP1.10026.001")
```

getProductSensors *Get data product-sensor relationships*

Description

Pull all data from the NEON API /products endpoint, create a data frame with data product ID, data product name, and sensor type.

Usage

```
getProductSensors()
```

Value

A data frame

Author(s)

Christine Laney <claney@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

```
## Not run:
sensors <- getProductSensors()

## End(Not run)
```

getReadmePublicationDate *Scrape the publication date from each ReadMe file*

Description

Given a directory, this will recursively list all of the ReadMe files that were unzipped. This should result in a single text file with a list of all of the publication dates from the ReadMe file.

Usage

```
getReadmePublicationDate(savepath, out_filepath, dpID)
```

Arguments

savepath	The root folder directory where the ReadMe files are located.
out_filepath	The output directory and filename.
dpID	The data product identifier

Author(s)

Nathan Mietkiewicz <mietkiewicz@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

---

getRecentPublication *Returns the most recent files for those that do not need stacking*

Description

Given a list of files, this will order and return the file with the most recent publication date.

Usage

```
getRecentPublication(inList)
```

Arguments

inList	The list of files.

Author(s)

Nathan Mietkiewicz <mietkiewicz@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

getTaxonTable *Get NEON taxon table*

Description

This is a function to retrieve a taxon table from the NEON data portal for the taxon type by the enduser.

Usage

```
getTaxonTable(
  taxonType = NA,
  recordReturnLimit = NA,
  stream = "true",
  token = NA
)
```

Arguments

taxonType	Character string for the taxonTypeCode. Must be one of ALGAE, BEETLE, BIRD, FISH, HERPETOLOGY, MACROINVERTEBRATE, MOSQUITO, MOSQUITO_PATHOGENS, SMALL_MAMMAL, PLANT, TICK
recordReturnLimit	Integer. The number of items to limit the result set to. If NA, will return all records in table.
stream	Character string, true or false. Option to obtain the result as a stream. Utilize for large requests.
token	User specific API token (generated within neon.datascience user accounts)

Value

data frame with selected NEON data

Author(s)

Eric R. Sokol <esokol@battelleecology.org>

References

getTileUrls — *Get and store the file names, S3 URLs, file size, and download status (default = 0) in a data frame*

Description

Produces a data frame that is populated by available tiles for the AOP product.

Usage

```
getTileUrls(m.urls, tileEasting, tileNorthing, token = NA_character_)
```

Arguments

m.urls	The monthly API URL for the AOP tile.
tileEasting	A vector containing the easting UTM coordinates of the locations to download.
tileNorthing	A vector containing the northing UTM coordinates of the locations to download.
token	User specific API token (generated within neon.datascience user accounts). Optional.

Value

A dataframe comprised of file names, S3 URLs, file size, and download status (default = 0)

Author(s)

Claire Lunch <clunch@battelleecology.org> Christine Laney <claney@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

getTimeIndex — *Get a list of the available time intervals for a data product*

Description

Most IS products are available at multiple time intervals; get a list of what's available for a given data product

Usage

```
getTimeIndex(dpID, token = NA_character_)
```

Arguments

dpID	The identifier of the NEON data product, in the form DPL.PRNUM.REV, e.g. DP1.00006.001
token	User specific API token (generated within neon.datascience user accounts)

Value

A vector of the available time intervals, typically in minutes.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

```
# Get available time intervals for PAR data product
getTimeIndex("DP1.00024.001")
```

---

getVarsEddy *Extract list of eddy covariance tables from HDF5 files*

Description

Extracts a list of table metadata from a single HDF5 file. Specific to eddy covariance data product: DP4.00200.001. Can inform inputs to stackEddy(); variables listed in 'name' are available inputs to the 'var' parameter in stackEddy().

Usage

```
getVarsEddy(filepath)
```

Arguments

filepath	The folder containing the H5 file [character]

Value

A data frame of the metadata for each data table in the HDF5 file

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

```
## Not run:
# read variables from a file in a hypothetical filepath
ec.vars <- getVarsEddy(filepath='/data/NEON.D19.BONA.DP4.00200.001.nsae.2017-12.basic.h5')

## End(Not run)
```

`getZipUrls` *Get and store the file names, S3 URLs, file size, and download status (default = 0) in a data frame*

Description

Used to generate a data frame of available zipfile URLs.

Usage

```
getZipUrls(
  month.urls,
  avg,
  package,
  dpID,
  release,
  messages,
  tabl,
  token = NA_character_
)
```

Arguments

`month.urls`	The monthly API URL for the URL files
`avg`	Global variable for averaging interval
`package`	Global varaible for package type (basic or expanded)
`dpID`	Global variable for data product ID
`release`	Data release to be downloaded
`messages`	Error/warning messages from previous steps
`tabl`	Table name to get
`token`	User specific API token (generated within neon.datascience user accounts)

Value

A dataframe comprised of file names, S3 URLs, file size, and download status (default = 0)

Author(s)

Claire Lunch <clunch@battelleecology.org> Christine Laney <claney@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

---

listFilesInZip *Get a data frame with the names of all files within a zipped NEON data package*

Description

Given the top level zip file, return dataframe of all of the files within it without unzipping the file

Usage

```
listFilesInZip(zippath)
```

Arguments

zippath	The path to a zip file

Value

A list of filenames within the given zip file

Author(s)

Christine Laney <claney@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

`listZipfiles` *Get all zip file names within a zipped NEON data package*

Description

Given the data frame of all the files within the top level zip file, return an array of just the zip file names (no pdf, xml, or other files).

Usage

```
listZipfiles(zippath)
```

Arguments

`zippath`	The path to a zip file

Value

An array of all zip files contained within the focal zip file

Author(s)

Christine Laney <claney@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

`loadByProduct` *Get files from NEON API, stack tables, and load into the current environment*

Description

Pull files from the NEON API, by data product, merge data for each table, and read into the current R environment

Usage

```
loadByProduct(
  dpID,
  site = "all",
  startdate = NA,
  enddate = NA,
  package = "basic",
  release = "current",
```

```
  timeIndex = "all",
  tabl = "all",
  check.size = TRUE,
  nCores = 1,
  forceParallel = FALSE,
  token = NA_character_,
  avg = NA
)
```

Arguments

dpID	The identifier of the NEON data product to pull, in the form DPL.PRNUM.REV, e.g. DP1.10023.001
site	Either the string 'all', meaning all available sites, or a character vector of 4-letter NEON site codes, e.g. c('ONAQ','RMNP'). Defaults to all.
startdate	Either NA, meaning all available dates, or a character vector in the form YYYY-MM, e.g. 2017-01. Defaults to NA.
enddate	Either NA, meaning all available dates, or a character vector in the form YYYY-MM, e.g. 2017-01. Defaults to NA.
package	Either 'basic' or 'expanded', indicating which data package to download. Defaults to basic.
release	The data release to be downloaded; either 'current' or the name of a release, e.g. 'RELEASE-2021'. 'current' returns provisional data in addition to the most recent release. To download only provisional data, use release='PROVISIONAL'. Defaults to 'current'.
timeIndex	Either the string 'all', or the time index of data to download, in minutes. Only applicable to sensor (IS) data. Defaults to 'all'.
tabl	Either the string 'all', or the name of a single data table to download. Defaults to 'all'.
check.size	T or F, should the user approve the total file size before downloading? Defaults to T. When working in batch mode, or other non-interactive workflow, use check.size=F.
nCores	The number of cores to parallelize the stacking procedure. By default it is set to a single core.
forceParallel	If the data volume to be processed does not meet minimum requirements to run in parallel, this overrides. Set to FALSE as default.
token	User specific API token (generated within neon.datascience user accounts)
avg	Deprecated; use timeIndex

Details

All available data meeting the query criteria will be downloaded. Most data products are collected at only a subset of sites, and dates of collection vary. Consult the NEON data portal for sampling details. Dates are specified only to the month because NEON data are provided in monthly packages. Any month included in the search criteria will be included in the download. Start and end date are inclusive.

Value

A named list of all the data tables in the data product downloaded, plus a validation file and a variables file, as available.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

```
## Not run:
# To download plant foliar properties data from all sites, expanded data package:
cfc <- loadByProduct(dpID="DP1.10026.001", site="all", package="expanded")

## End(Not run)
```

---

other_bundles *Bundled vegetation and sediment data product information*

Description

A dataset containing NEON data product codes of vegetation and sediment data products and the "home" data products they are bundled with.

Usage

```
other_bundles
```

Format

A data frame with 2 variables:

product Data product ID of a product

homeProduct Data product ID of the corresponding home data product

Source

NEON data product bundles

`quietMessages` *Will suppress all output messages, while retaining the output dataframe*

Description

Used to quiet all output messages

Usage

```
quietMessages(toBeQuieted)
```

Arguments

`toBeQuieted`	Input to be quieted

Value

The expected output without associated messages/warnings.

Author(s)

Nate Mietkiewicz <mietkiewicz@battelleecology.org>

References

`readTableNEON` *Read a NEON data table with correct data types for each variable*

Description

Load a table into R, assigning classes to each column based on data types in variables file; or convert a table already loaded

Usage

```
readTableNEON(dataFile, varFile)
```

Arguments

`dataFile`	A data frame containing a NEON data table, or the filepath to a data table to load
`varFile`	A data frame containing the corresponding NEON variables file, or the filepath to the variables file

Value

A data frame of a NEON data table, with column classes assigned by data type

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

`shared_aquatic` *Terrestrial-aquatic shared data information*

Description

A dataset containing NEON site codes and data product IDs for places where meteorological data from terrestrial sites are used as the data of record for nearby aquatic sites as well.

Usage

```
shared_aquatic
```

Format

A data frame with 3 variables:

site site code of a NEON aquatic site

towerSite site code of the NEON terrestrial site used as the data source for the corresponding aquatic site

product Data product ID of the data products to which the corresponding terrestrial-aquatic site relationship relates

Source

NEON site layouts and spatial design

shared_flights *Flight coverage information*

Description

A dataset containing NEON site codes for places where a single AOP flight may cover multiple sites

Usage

```
shared_flights
```

Format

A data frame with 2 variables:

site site code of a NEON site

flightSite site code that matches the file naming for flights that may include "site"

Source

NEON flight plans

stackByTable *Join data files in a zipped NEON data package by table type*

Description

Given a zipped data file, do a full join of all data files, grouped by table type. This should result in a small number of large files.

Usage

```
stackByTable(
  filepath,
  savepath = NA,
  folder = FALSE,
  saveUnzippedFiles = FALSE,
  dpID = NA,
  package = NA,
  nCores = 1
)
```

Arguments

filepath	The location of the zip file
savepath	The location to save the output files to
folder	T or F: does the filepath point to a parent, unzipped folder, or a zip file? If F, assumes the filepath points to a zip file. Defaults to F. No longer needed; included for back compatibility.
saveUnzippedFiles	T or F: should the unzipped monthly data folders be retained?
dpID	Data product ID of product to stack. Ignored and determined from data unless input is a vector of files, generally via stackFromStore().
package	Data download package, either basic or expanded. Ignored and determined from data unless input is a vector of files, generally via stackFromStore().
nCores	The number of cores to parallelize the stacking procedure. To automatically use the maximum number of cores on your machine we suggest setting nCores=parallel::detectCores(). By default it is set to a single core.

Value

All files are unzipped and one file for each table type is created and written. If savepath="envt" is specified, output is a named list of tables; otherwise, function output is null and files are saved to the location specified.

Author(s)

Christine Laney <claney@battelleecology.org> Claire Lunch <clunch@battelleecology.org>

References

Examples

```
## Not run:
# To unzip and merge files downloaded from the NEON Data Portal
stackByTable("~/NEON_par.zip")

# To unzip and merge files downloaded using zipsByProduct()
stackByTable("~/filesToStack00024")

## End(Not run)
```

stackDataFilesParallel *Join data files in a unzipped NEON data package by table type*

Description

Given a folder of unzipped files (unzipped NEON data file), do a full join of all data files, grouped by table type. This should result in a small number of large files.

Usage

```
stackDataFilesParallel(folder, nCores = 1, dpID)
```

Arguments

folder	The location of the data
nCores	The number of cores to parallelize the stacking procedure. To automatically use the maximum number of cores on your machine we suggest setting 'nCores=parallel::detectCores()'. By default it is set to a single core. If the files are less than 25000 bytes the userdefined nCores will be overridden to a single core.
dpID	The data product identifier

Value

One file for each table type is created and written.

Author(s)

Christine Laney <claney@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

stackEddy *Extract eddy covariance data from HDF5 format*

Description

Convert data of choice from HDF5 to tabular format. Specific to eddy covariance data product: DP4.00200.001

Usage

```
stackEddy(filepath, level = "dp04", var = NA, avg = NA)
```

Arguments

filepath	One of: a folder containing NEON EC H5 files, a zip file of DP4.00200.001 data downloaded from the NEON data portal, a folder of DP4.00200.001 data downloaded by the neonUtilities::zipsByProduct() function, or a single NEON EC H5 file [character]
level	The level of data to extract; one of dp01, dp02, dp03, dp04 [character]
var	The variable set to extract. Can be any of the variables in the "name" level or the "system" level of the H5 file; use the getVarsEddy() function to see the available variables. From the inputs, all variables from "name" and all variables from "system" will be returned, but if variables from both "name" and "system" are specified, the function will return only the intersecting set. This allows the user to, e.g., return only the pressure data ("pres") from the CO2 storage system ("co2Stor"), instead of all the pressure data from all instruments. [character]
avg	The averaging interval to extract, in minutes [numeric]

Details

Given a filepath containing H5 files of DP4.00200.001 data, extracts variables, stacks data tables over time, and joins variables into a single table. For data product levels 2-4 (dp02, dp03, dp04), joins all available data, except for the flux footprint data in the expanded package. For dp01, an averaging interval and a set of variable names must be provided as inputs.

Value

A named list of data frames. One data frame per site, plus one data frame containing the metadata (objDesc) table and one data frame containing units for each variable (variables).

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

Examples

```
## Not run:
# To extract and merge Level 4 data tables, where data files are in the working directory
flux <- stackEddy(filepath=getwd(), level='dp04', var=NA, avg=NA)

## End(Not run)
```

`stackFromStore` *Select files from a stored set of NEON data, created by neonstore package methods or another method*

Description

Select files from a stored set based on input criteria and pass to stackByTable() or stackEddy()

Usage

```
stackFromStore(
  filepaths,
  dpID,
  site = "all",
  startdate = NA,
  enddate = NA,
  pubdate = NA,
  timeIndex = "all",
  level = "dp04",
  var = NA,
  zipped = FALSE,
  package = "basic",
  load = TRUE,
  nCores = 1
)
```

Arguments

`filepaths`	Either a vector of filepaths pointing to files to be stacked, or a single directory containing files that can be stacked, with selection criteria detmined by the other inputs. In both cases files to be stacked must be either site-month zip files or unzipped folders corresponding to site-month zips. [character]
`dpID`	The NEON data product ID of the data to be stacked [character]
`site`	Either "all" or a vector of NEON site codes to be stacked [character]
`startdate`	Either NA, meaning all available dates, or a character vector in the form YYYY-MM, e.g. 2017-01. Defaults to NA. [character]
`enddate`	Either NA, meaning all available dates, or a character vector in the form YYYY-MM, e.g. 2017-01. Defaults to NA. [character]
`pubdate`	The maximum publication date of data to include in stacking, in the form YYYY-MM-DD. If NA, the most recently published data for each product-site-month combination will be selected. Otherwise, the most recent publication date that is older than pubdate will be selected. Thus the data stacked will be the data that would have been accessed on the NEON Data Portal, if it had been downloaded on pubdate. [character]
`timeIndex`	Either the string 'all', or the time index of data to be stacked, in minutes. Only applicable to sensor (IS) and eddy covariance data. Defaults to 'all'. [character]

level	Data product level of data to stack. Only applicable to eddy covariance (SAE) data; see stackEddy() documentation. [character]
var	Variables to be extracted and stacked from H5 files. Only applicable to eddy covariance (SAE) data; see stackEddy() documentation. [character]
zipped	Should stacking use data from zipped files or unzipped folders? This option allows zips and their equivalent unzipped folders to be stored in the same directory; stacking will extract whichever is specified. Defaults to FALSE, i.e. stacking using unzipped folders. [logical]
package	Either "basic" or "expanded", indicating which data package to stack. Defaults to basic. [character]
load	If TRUE, stacked data are read into the current R environment. If FALSE, stacked data are written to the directory where data files are stored. Defaults to TRUE. [logical]
nCores	Number of cores to use for optional parallel processing. Defaults to 1. [integer]

Value

If load=TRUE, returns a named list of stacked data tables. If load=FALSE, return is empty and stacked files are written to data directory.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

---

table_types *Publication table information*

Description

A dataset containing publication table names, descriptions, type (site-date, site-all, lab-all, lab-current), and a time index

Usage

```
table_types
```

Format

A data frame with 5 variables. Number of rows changes frequently as more tables are added:

productID data product ID

tableName name of table

tableDesc description of table

tableType type of table (important for knowing which tables to stack, and how to stack)

tableTMI a time index (e.g., 0 = native resolution, 1 = 1 minute, 30 = 30 minute averages or totals)

Source

NEON database

---

transformFileToGeoCSV *Transform NEON CSV file to GeoCSV*

Description

Read in a NEON monthly data zip file and parse the respective variables file to create a new GeoCSV file

Usage

```
transformFileToGeoCSV(infile, varfile, outfile)
```

Arguments

infile	The path to the file that needs to be parsed
varfile	The path to the variables file needed to parse the infile
outfile	The path where the new GeoCSV file should be placed

Value

The same data file with a GeoCSV header

Author(s)

Christine Laney <claney@battelleecology.org>

References

unzipZipfileParallel *Unzip a zip file either at just the top level or recursively through the file*

Description

Unzip a zip file either at just the top level or recursively through the file

Usage

```
unzipZipfileParallel(
  zippath,
  outpath = substr(zippath, 1, nchar(zippath) - 4),
  level = "all",
  nCores = 1
)
```

Arguments

zippath	The filepath of the input file
outpath	The name of the folder to save unpacked files to
level	Whether the unzipping should occur only for the 'top' zip file, or unzip 'all' recursively, or only files 'in' the folder specified
nCores	Number of cores to use for parallelization

Author(s)

Christine Laney <claney@battelleecology.org> Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

zipsByProduct *Get files from NEON API to feed the stackByTable() function*

Description

Pull files from the NEON API, by data product, in a structure that will allow them to be stacked by the stackByTable() function

Usage

```
zipsByProduct(
  dpID,
  site = "all",
  startdate = NA,
  enddate = NA,
  package = "basic",
  release = "current",
  timeIndex = "all",
  tabl = "all",
  check.size = TRUE,
  savepath = NA,
  load = F,
  token = NA_character_,
  avg = NA
)
```

Arguments

dpID	The identifier of the NEON data product to pull, in the form DPL.PRNUM.REV, e.g. DP1.10023.001
site	Either the string 'all', meaning all available sites, or a character vector of 4-letter NEON site codes, e.g. c('ONAQ','RMNP'). Defaults to all.
startdate	Either NA, meaning all available dates, or a character vector in the form YYYY-MM, e.g. 2017-01. Defaults to NA.
enddate	Either NA, meaning all available dates, or a character vector in the form YYYY-MM, e.g. 2017-01. Defaults to NA.
package	Either 'basic' or 'expanded', indicating which data package to download. Defaults to basic.
release	The data release to be downloaded; either 'current' or the name of a release, e.g. 'RELEASE-2021'. 'current' returns provisional data in addition to the most recent release. To download only provisional data, use release='PROVISIONAL'. Defaults to 'current'.
timeIndex	Either the string 'all', or the time index of data to download, in minutes. Only applicable to sensor (IS) data. Defaults to 'all'.
tabl	Either the string 'all', or the name of a single data table to download. Defaults to 'all'.
check.size	T or F, should the user approve the total file size before downloading? Defaults to T. When working in batch mode, or other non-interactive workflow, use check.size=F.
savepath	The location to save the output files to
load	T or F, are files saved locally or loaded directly? Used silently with loadByProduct(), do not set manually.
token	User specific API token (generated within neon.datascience user accounts). Optional.
avg	Deprecated; use timeIndex

Details

All available data meeting the query criteria will be downloaded. Most data products are collected at only a subset of sites, and dates of collection vary. Consult the NEON data portal for sampling details. Dates are specified only to the month because NEON data are provided in monthly packages. Any month included in the search criteria will be included in the download. Start and end date are inclusive. timeIndex: NEON sensor data are published at pre-determined averaging intervals, usually 1 and 30 minutes. The default download includes all available data. Download volume can be greatly reduced by downloading only the 30 minute files, if higher frequency data are not needed. Use getTimeIndex() to find the available averaging intervals for each sensor data product.

Value

A folder in the working directory (or in savepath, if specified), containing all zip files meeting query criteria.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

```
## Not run:
# To download plant foliar properties data from all sites, expanded data package:
zipsByProduct(dpID="DP1.10026.001", site="all", package="expanded")

## End(Not run)
```

---

zipsByURI *Get files from NEON ECS Bucket using URLs in stacked data*

Description

Read in a set of URLs from NEON data tables and then download the data from the NEON ECS buckets. Assumes data tables are in the format resulting from merging files using stackByTable(). File downloads from ECS can be extremely large; be prepared for long download times and large file storage.

Usage

```
zipsByURI(
  filepath,
  savepath = paste0(filepath, "/ECS_zipFiles"),
  pick.files = FALSE,
  check.size = TRUE,
```

```
  unzip = TRUE,
  saveZippedFiles = FALSE
)
```

Arguments

filepath	The location of the NEON data containing URIs. Can be either a local directory containing NEON tabular data or a list object containing tabular data.
savepath	The location to save the output files from the ECS bucket, optional. Defaults to creating a "ECS_zipFiles" folder in the filepath directory.
pick.files	T or F, should the user be told the name of each file before downloading? Defaults to F. When working in batch mode, or other non-interactive workflow, use pick.files=F.
check.size	T or F, should the user be told the total file size before downloading? Defaults to T. When working in batch mode, or other non-interactive workflow, use check.size=F.
unzip	T or F, indicates if the downloaded zip files from ECS buckets should be unzipped into the same directory, defaults to T. Supports .zip and .tar.gz files currently.
saveZippedFiles	T or F: should the zip files be retained after unzipping? Defaults to F.

Value

A folder in the working directory (or in savepath, if specified), containing all files meeting query criteria.

Author(s)

Kaelin Cawley <kcawley@battelleecology.org>

References

Examples

```
## Not run:
# To download stream morphology data from stacked data:
zipsByURI(filepath="~/filesToStack00131/stackedFiles")

## End(Not run)
```

Index