Package 'nhdplusTools'

November 7, 2019

Type Package

Title NHDPlus Tools

Version 0.3.11

Description Tools for traversing and working with National Hydrography Dataset Plus (NHDPlus) data. All methods implemented in 'nhdplusTools' are available in the NHDPlus documentation available from the US Environmental Protection Agency <https://www.epa.gov/waterdata/basic-information>.

URL https://github.com/usgs-r/nhdplusTools

BugReports https://github.com/usgs-r/nhdplusTools/issues

Depends R (>= 3.5.0)

Imports dplyr, sf, RANN, units, magrittr, jsonlite, httr, igraph, xml2, R.utils, utils

Suggests testthat, knitr, rmarkdown, rosm, prettymapr, ggmap, ggplot2, sp, lwgeom, devtools, codetools

License CC0

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author David Blodgett [aut, cre], Mike Johnson [ctb]

Maintainer David Blodgett <dblodgett@usgs.gov>

Repository CRAN

Date/Publication 2019-11-07 14:50:02 UTC

R topics documented:

`align_nhdplus_names` *Align NHD Dataset Names*

Description

this function takes any NHDPlus dataset and aligns the attribute names with those used in nhdplusTools.

Usage

```
align_nhdplus_names(x)
```

Arguments

`x`	a `sf` object of nhdplus flowlines

Value

a renamed `sf` object

Examples

```
source(system.file("extdata/new_hope_data.R", package = "nhdplusTools"))

names(new_hope_flowline)

names(new_hope_flowline) <- tolower(names(new_hope_flowline))

new_hope_flowline <- align_nhdplus_names(new_hope_flowline)

names(new_hope_flowline)
```

`calculate_arbolate_sum` *Calculate Arbolate Sum*

Description

Calculates arbolate sum given a dendritic network and incremental lengths. Arbolate sum is the total length of all upstream flowlines.

Usage

```
calculate_arbolate_sum(catchment_area)
```

Arguments

`catchment_area`	data.frame with ID, toID, and length columns.

Value

numeric with arbolate sum.

Examples

```
library(dplyr)
source(system.file("extdata", "walker_data.R", package = "nhdplusTools"))
catchment_length <- prepare_nhdplus(walker_flowline, 0, 0,
                             purge_non_dendritic = FALSE, warn = FALSE) %>%
  left_join(select(walker_flowline, COMID), by = "COMID") %>%
  select(ID = COMID, toID = toCOMID, length = LENGTHKM)

arb_sum <- calculate_arbolate_sum(catchment_length)

catchment_length$arb_sum <- arb_sum
catchment_length$nhd_arb_sum <- walker_flowline$ArbolateSu

mean(abs(catchment_length$arb_sum - catchment_length$nhd_arb_sum))
max(abs(catchment_length$arb_sum - catchment_length$nhd_arb_sum))
```

`calculate_levelpaths` *Calculate Level Paths*

Description

Calculates level paths using the stream-leveling approach of NHD and NHDPlus. In addition to a levelpath identifier, a topological sort and levelpath outlet identifier is provided in output. If arbolate sum is provided in the weight column, this will match the behavior of NHDPlus. Any numeric value can be included in this column and the largest value will be followed when no nameID is available.

Usage

```
calculate_levelpaths(flowline, status = FALSE)
```

Arguments

`flowline`	data.frame with ID, toID, nameID, and weight columns.
`status`	boolean if status updates should be printed.

Details

1. levelpath provides an identifier for the collection of flowlines that make up the single mainstem flowpath of a total upstream aggregate catchment.
2. outletID is the catchment ID (COMID in the case of NHDPlus) for the catchment at the outlet of the levelpath the catchment is part of.
3. topo_sort is similar to Hydroseq in NHDPlus in that large topo_sort values are upstream of small topo_sort values. Note that there are many valid topological sort orders of a directed graph. The sort order output by this function is generated using 'igraph::topo_sort'.

Value

data.frame with ID, outletID, topo_sort, and levelpath collumns. See details for more info.

Examples

```
source(system.file("extdata", "walker_data.R", package = "nhdplusTools"))

test_flowline <- prepare_nhdplus(walker_flowline, 0, 0, FALSE)

test_flowline <- data.frame(
  ID = test_flowline$COMID,
  toID = test_flowline$toCOMID,
  nameID = walker_flowline$GNIS_ID,
  weight = walker_flowline$ArbolateSu,
  stringsAsFactors = FALSE)

calculate_levelpaths(test_flowline)
```

calculate_total_drainage_area *Total Drainage Area*

Description

Calculates total drainage area given a dendritic network and incremental areas.

Usage

```
calculate_total_drainage_area(catchment_area)
```

Arguments

catchment_area	data.frame with ID, toID, and area columns.

Value

numeric with total area.

Examples

```
library(dplyr)
source(system.file("extdata", "walker_data.R", package = "nhdplusTools"))
catchment_area <- prepare_nhdplus(walker_flowline, 0, 0,
                                  purge_non_dendritic = FALSE, warn = FALSE) %>%
  left_join(select(walker_flowline, COMID, AreaSqKM), by = "COMID") %>%
  select(ID = COMID, toID = toCOMID, area = AreaSqKM)

new_da <- calculate_total_drainage_area(catchment_area)

catchment_area$totda <- new_da
catchment_area$nhdptotda <- walker_flowline$TotDASqKM

mean(abs(catchment_area$totda - catchment_area$nhdptotda))
max(abs(catchment_area$totda - catchment_area$nhdptotda))
```

discover_nhdplus_id *Discover NHDPlus ID*

Description

Multipurpose function to find a COMID of interest.

Usage

```
discover_nhdplus_id(point = NULL, nldi_feature = NULL)
```

Arguments

point	An sf POINT including crs as created by: sf::st_sfc(sf::st_point(..,..), crs)
nldi_feature	list with names ‘featureSource‘ and ‘featureID‘ where ‘featureSource‘ is derived from the "source" column of the response of discover_nldi_sources() and the ‘featureSource‘ is a known identifier from the specified ‘featureSource‘.

Value

integer COMID

Examples

```
point <- sf::st_sfc(sf::st_point(c(-76.87479, 39.48233)), crs = 4326)
discover_nhdplus_id(point)

nldi_nwis <- list(featureSource = "nwissite", featureID = "USGS-08279500")
discover_nhdplus_id(nldi_feature = nldi_nwis)
```

---

discover_nldi_navigation

Discover NLDI Navigation Options

Description

Discover available navigation options for a given feature source and id.

Usage

```
discover_nldi_navigation(nldi_feature, tier = "prod")
```

Arguments

nldi_feature	list with names ‘featureSource‘ and ‘featureID‘ where ‘featureSource‘ is derived from the "source" column of the response of discover_nldi_sources() and the ‘featureSource‘ is a known identifier from the specified ‘featureSource‘.
tier	character optional "prod" or "test"

Value

data.frame with three columns "source", "sourceName" and "features"

Examples

```
discover_nldi_sources()

nldi_nwis <- list(featureSource = "nwissite", featureID = "USGS-08279500")

discover_nldi_navigation(nldi_nwis)
```

---

discover_nldi_sources *Discover NLDI Sources*

Description

Function to retrieve available feature and data sources from the Network Linked Data Index.

Usage

```
discover_nldi_sources(tier = "prod")
```

Arguments

tier	character optional "prod" or "test"

Value

data.frame with three columns "source", "sourceName" and "features"

Examples

```
discover_nldi_sources()
```

---

download_nhdplushr *Download NHDPlus HiRes*

Description

Download NHDPlus HiRes

Usage

```
download_nhdplushr(nhd_dir, hu_list, download_files = TRUE)
```

Arguments

nhd_dir	character directory to save output into
hu_list	character vector of hydrologic region(s) to download
download_files	boolean if FALSE, only URLs to files will be returned can be hu02s and/or hu04s

Value

Paths to geodatabases created.

Examples

```
download_nhdplushr(tempdir(), c("01", "0203"), download_files = FALSE)
```

---

download_nhdplusv2 *Download seamless National Hydrography Dataset Version 2 (NHDPlusV2)*

Description

This function downloads and decompresses staged seamless NHDPlusV2 data. The following requirements are needed: p7zip (MacOS), 7zip (windows) Please see: https://www.epa.gov/waterdata/get-nhdplus-national-hydrography-dataset-plus-data for more information and metadata about this data.

Usage

```
download_nhdplusv2(outdir,
  url = paste0("https://s3.amazonaws.com/nhdplus/NHDPlusV21/",
  "Data/NationalData/NHDPlusV21_NationalData_Seamless",
  "_Geodatabase_Lower48_07.7z"))
```

Arguments

outdir	The folder path where data should be downloaded and extracted
url	the location of the online resource

Value

the path to the local geodatabase

Examples

```
## Not run:
  download_nhdplusV2("./data/nhd/")

## End(Not run)
```

download_rf1 *Download the seamless Reach File (RF1) Database*

Description

This function downloads and decompresses staged RF1 data. See: https://water.usgs.gov/GIS/metadata/usgswrd/XML/erf1_2
for metadata.

Usage

```
download_rf1(outdir,
  url = "https://water.usgs.gov/GIS/dsdl/erf1_2.e00.gz")
```

Arguments

outdir	The folder path where data should be downloaded and extracted
url	the location of the online resource

Value

the path to the local e00 file

Examples

```
## Not run:
  download_wbd("./data/rf1/")

## End(Not run)
```

download_wbd *Download the seamless Watershed Boundary Dataset (WBD)*

Description

This function downloads and decompresses staged seamless WBD data. Please see: https://prd-tnm.s3.amazonaws.com/StagedProducts/Hydrography/WBD/National/GDB/WBD_National_GDB.xml
for metadata.

Usage

```
download_wbd(outdir,
  url = paste0("https://prd-tnm.s3.amazonaws.com/StagedProducts/",
  "Hydrography/WBD/National/GDB/WBD_National_GDB.zip"))
```

Arguments

outdir	The folder path where data should be downloaded and extracted
url	the location of the online resource

Value

the path to the local geodatabase

Examples

```
## Not run:
  download_wbd("./data/wbd/")

## End(Not run)
```

---

get_DD *Navigate Downstream with Diversions*

Description

Traverse NHDPlus network downstream with diversions NOTE: This algorithm may not scale well in large watersheds. For reference, the lower Mississippi will take over a minute.

Usage

```
get_DD(network, comid, distance = NULL)
```

Arguments

network	data.frame NHDPlus flowlines including at a minimum: COMID, DnMinorHyd, DnHydroseq, and Hydroseq.
comid	integer identifier to start navigating from.
distance	numeric distance in km to limit how many COMIDs are returned. The COMID that exceeds the distance specified is returned. The longest of the diverted paths is used for limiting distance.

Value

integer vector of all COMIDs downstream of the starting COMID

Examples

```
library(sf)
start_COMID <- 11688818
sample_flines <- read_sf(system.file("extdata",
                                     "petapsco_flowlines.gpkg",
                                     package = "nhdplusTools"))
DD_COMIDs <- get_DD(sample_flines, start_COMID, distance = 4)
plot(dplyr::filter(sample_flines, COMID %in% DD_COMIDs)$geom,
     col = "red", lwd = 2)

DM_COMIDs <- get_DM(sample_flines, start_COMID, distance = 4)
plot(dplyr::filter(sample_flines, COMID %in% DM_COMIDs)$geom,
     col = "blue", add = TRUE, lwd = 2)
```

---

get_DM *Navigate Downstream Mainstem*

Description

Traverse NHDPlus network downstream main stem

Usage

```
get_DM(network, comid, distance = NULL, sort = FALSE, include = TRUE)
```

Arguments

network	data.frame NHDPlus flowlines including at a minimum: COMID, LENGTHKM, DnHydroseq, and Hydroseq.
comid	integer identifier to start navigating from.
distance	numeric distance in km to limit how many COMIDs are returned. The COMID that exceeds the distance specified is returned.
sort	if TRUE, the returned COMID vector witll be sorted in order of distance from the input COMID (nearest to farthest)
include	if TRUE, the input COMID will be included in the returned COMID vector

Value

integer vector of all COMIDs downstream of the starting COMID along the mainstem

Examples

```
library(sf)
sample_flines <- read_sf(system.file("extdata",
                                     "petapsco_flowlines.gpkg",
                                     package = "nhdplusTools"))
plot(sample_flines$geom)
start_COMID <- 11690092
DM_COMIDs <- get_DM(sample_flines, start_COMID)
plot(dplyr::filter(sample_flines, COMID %in% DM_COMIDs)$geom,
     col = "red", add = TRUE, lwd = 3)

DM_COMIDs <- get_DM(sample_flines, start_COMID, distance = 40)
plot(dplyr::filter(sample_flines, COMID %in% DM_COMIDs)$geom,
     col = "blue", add = TRUE, lwd = 2)
```

---

get_flowline_index *Get Flowline Index*

Description

given an sf point geometry column, return COMID, reachcode, and measure for each.

Usage

```
get_flowline_index(flines, points, search_radius = 0.1, precision = NA)
```

Arguments

flines	sf data.frame of type LINESTRING or MULTILINESTRING including COMID, REACHCODE, ToMeas, and FromMeas
points	sfc of type POINT
search_radius	numeric the distance for the nearest neighbor search to extend. See RANN nn2 documentation for more details.
precision	numeric the resolution of measure precision in the output.

Details

Note 1: Inputs are cast into LINESTRINGS. Because of this, the measure output of inputs that are true multipart lines may be in error.

Note 2: This algorithm finds the nearest node in the input flowlines to identify which flowline the point should belong to. As a second pass, it can calculate the measure to greater precision than the nearest flowline geometry node.

Note 3: Offset is returned in units consistant with the projection of the flowlines.

Value

data.frame with four columns, COMID, REACHCODE, REACH_meas, and offset.

Examples

```
sample_flines <- sf::read_sf(system.file("extdata",
                                         "petapsco_flowlines.gpkg",
                                         package = "nhdplusTools"))
get_flowline_index(sample_flines,
                   sf::st_sfc(sf::st_point(c(-76.87479,
                                             39.48233)),
                              crs = 4326))
```

`get_nhdplushr` *Get NHDPlus HiRes as single geopackage*

Description

Get NHDPlus HiRes as single geopackage

Usage

```
get_nhdplushr(hr_dir, out_gpkg = NULL, layers = c("NHDFlowline",
  "NHDPlusCatchment"), pattern = ".*GDB.gdb$")
```

Arguments

`hr_dir`	character directory with geodatabases (gdb search is recursive)
`out_gpkg`	character path to write output geopackage
`layers`	character vector with desired layers to return. c("NHDFlowline", "NHDPlusCatchment") is default. Choose from: c("NHDFlowline", "NHDPlusCatchment", "NHDWaterbody", "NHDArea", "NHDLine", "NHDPlusSink", "NHDPlusWall", "NHDPoint", "NHDPlusBurnWaterbody", "NHDPlusBurnLineEvent", "HYDRO_NET_Junctions", "WBDHU2", "WBDHU4","WBDHU6", "WBDHU8" "WBDHU10", "WBDHU12", "WBDLine")
`pattern`	character optional regex to select certain files in hr_dir

Details

NHDFlowline is joined to value added attributes prior to being returned. Names are not modified from the NHDPlusHR geodatabase. Set layers to "NULL" to get all layers.

Examples

```
# Note this will download a lot of data to a temp directory.
# Change 'tempdir()' to your directory of choice.
download_dir <- download_nhdplushr(tempdir(), c("0302", "0303"))
get_nhdplushr(download_dir, file.path(download_dir, "nhdplus_0302-03.gpkg"))
```

---

get_nldi_basin *Get NLDI Basin Boundary*

Description

Get a basin boundary for a given NLDI feature.

Usage

```
get_nldi_basin(nldi_feature, tier = "prod")
```

Arguments

nldi_feature	list with names ‘featureSource‘ and ‘featureID‘ where ‘featureSource‘ is derived from the "source" column of the response of discover_nldi_sources() and the ‘featureSource‘ is a known identifier from the specified ‘featureSource‘.
tier	character optional "prod" or "test"

Details

Only resolves to the nearest NHDPlus catchment divide. See: https://owi.usgs.gov/blog/nldi-intro/ for more info on the nldi.

Value

sf data.frame with result basin boundary

Examples

```
library(sf)
library(dplyr)

nldi_nwis <- list(featureSource = "nwissite", featureID = "USGS-05428500")

basin <- get_nldi_basin(nldi_feature = nldi_nwis)

basin %>%
 st_geometry() %>%
 plot()
```

basin

---

get_UM *Navigate Upstream Mainstem*

Description

Traverse NHDPlus network upstream main stem

Usage

```
get_UM(network, comid, distance = NULL, sort = FALSE, include = TRUE)
```

Arguments

network	data.frame NHDPlus flowlines including at a minimum: COMID,Pathlength, LevelPathI, UpHydroseq, and Hydroseq.
comid	integer identifier to start navigating from.
distance	numeric distance in km to limit how many COMIDs are
sort	if TRUE, the returned COMID vector witll be sorted in order of distance from the input COMID (nearest to farthest)
include	if TRUE, the input COMID will be included in the returned COMID vector returned. The COMID that exceeds the distance specified is returned.

Value

integer vector of all COMIDs upstream of the starting COMID along the mainstem

Examples

```
library(sf)
sample_flines <- read_sf(system.file("extdata",
                                     "petapsco_flowlines.gpkg",
                                     package = "nhdplusTools"))
plot(sample_flines$geom)
start_COMID <- 11690196
UM_COMIDs <- get_UM(sample_flines, start_COMID)
plot(dplyr::filter(sample_flines, COMID %in% UM_COMIDs)$geom,
     col = "red", add = TRUE, lwd = 3)

UM_COMIDs <- get_UM(sample_flines, start_COMID, distance = 50)
plot(dplyr::filter(sample_flines, COMID %in% UM_COMIDs)$geom,
     col = "blue", add = TRUE, lwd = 2)
```

get_UT *Navigate Upstream with Tributaries*

Description

Traverse NHDPlus network upstream with tributaries

Usage

```
get_UT(network, comid, distance = NULL)
```

Arguments

network	data.frame NHDPlus flowlines including at a minimum: COMID, Pathlength, LENGTHKM, and Hydroseq.
comid	integer Identifier to start navigating from.
distance	numeric distance in km to limit how many COMIDs are returned. The COMID that exceeds the distance specified is returned.

Value

integer vector of all COMIDs upstream with tributaries of the starting COMID.

Examples

```
library(sf)
sample_flines <- read_sf(system.file("extdata",
                                     "petapsco_flowlines.gpkg",
                                     package = "nhdplusTools"))
plot(sample_flines$geom)
start_COMID <- 11690196
UT_COMIDs <- get_UT(sample_flines, start_COMID)
plot(dplyr::filter(sample_flines, COMID %in% UT_COMIDs)$geom,
     col = "red", add = TRUE)

UT_COMIDs <- get_UT(sample_flines, start_COMID, distance = 50)
plot(dplyr::filter(sample_flines, COMID %in% UT_COMIDs)$geom,
     col = "blue", add = TRUE)
```

navigate_nldi *Navigate NLDI*

Description

Navigate the Network Linked Data Index network.

Usage

```
navigate_nldi(nldi_feature, mode = "upstreamMain",
  data_source = "comid", distance_km = NULL, tier = "prod")
```

Arguments

nldi_feature	list with names ‘featureSource‘ and ‘featureID‘ where ‘featureSource‘ is derived from the "source" column of the response of discover_nldi_sources() and the ‘featureSource‘ is a known identifier from the specified ‘featureSource‘.
mode	character chosen from names, URLs, or url parameters returned by discover_nldi_navigation(nldi_feature) See examples.
data_source	character chosen from "source" column of the response of discover_nldi_sources() or empty string for flowline geometry.
distance_km	numeric distance in km to stop navigating.
tier	character optional "prod" or "test"

Value

sf data.frame with result

Examples

```
library(sf)
library(dplyr)

nldi_nwis <- list(featureSource = "nwissite", featureID = "USGS-05428500")

navigate_nldi(nldi_feature = nldi_nwis,
              mode = "upstreamTributaries",
              data_source = "") %>%
  st_geometry() %>%
  plot()

navigate_nldi(nldi_feature = nldi_nwis,
              mode = "UM",
              data_source = "") %>%
  st_geometry() %>%
  plot(col = "blue", add = TRUE)
```

```
nwissite <- navigate_nldi(nldi_feature = nldi_nwis,
                          mode = "UT",
                          data_source = "nwissite")

st_geometry(nwissite) %>%
  plot(col = "green", add = TRUE)

nwissite
```

---

nhdplus_path *NHDPlus Data Path*

Description

Allows specification of a custom path to a source dataset. Typically this will be the national seamless dataset in geodatabase or geopackage format.

Usage

```
nhdplus_path(path = NULL, warn = FALSE)
```

Arguments

path	character path ending in .gdb or .gpkg
warn	boolean controls whether warning an status messages are printed

Value

1 if set successfully, the path if no input.

Examples

```
nhdplus_path("/data/NHDPlusV21_National_Seamless.gdb")

nhdplus_path("/data/NHDPlusV21_National_Seamless.gdb", warn=FALSE)

nhdplus_path()
```

prepare_nhdplus *Prep NHDPlus Data*

Description

Function to prep NHDPlus data for use by nhdplusTools functions

Usage

```
prepare_nhdplus(flines, min_network_size, min_path_length,
  min_path_size = 0, purge_non_dendritic = TRUE, warn = TRUE,
  error = TRUE)
```

Arguments

flines	data.frame NHDPlus flowlines including: COMID, LENGTHKM, FTYPE, TerminalFl, FromNode, ToNode, TotDASqKM, StartFlag, StreamOrde, StreamCalc, TerminalPa, Pathlength, and Divergence variables.
min_network_size	numeric Minimum size (sqkm) of drainage network to include in output.
min_path_length	numeric Minimum length (km) of terminal level path of a network.
min_path_size	numeric Minimum size (sqkm) of outlet level path of a drainage basin. Drainage basins with an outlet drainage area smaller than this will be removed.
purge_non_dendritic	boolean Should non dendritic paths be removed or not.
warn	boolean controls whether warning an status messages are printed
error	boolean controls whether to return potentially invalid data with a warning rather than an error

Value

data.frame ready to be used with the refactor_flowlines function.

Examples

```
flines_in <- sf::read_sf(system.file("extdata/petapsco_flowlines.gpkg",
                                     package = "nhdplusTools"))
prepare_nhdplus(flines_in,
                min_network_size = 10,
                min_path_length = 1,
                warn = FALSE)
```

`stage_national_data` *Stage NHDPlus National Data*

Description

Breaks down the national geo database into a collection of quick to access R binary files.

Usage

```
stage_national_data(include = c("attribute", "flowline", "catchment"),
  output_path = NULL, nhdplus_data = NULL, simplified = TRUE)
```

Arguments

`include`	character vector containing one or more of: "attributes", "flowline", "catchment".
`output_path`	character path to save the output to defaults to the directory of the nhdplus_data.
`nhdplus_data`	character path to the .gpkg or .gdb containing the national seamless dataset. Not required if `nhdplus_path` has been set.
`simplified`	boolean if TRUE (the default) the CatchmentSP layer will be included.

Details

"attributes" will save 'NHDFlowline_Network' attributes as a seperate data.frame without the geometry. The others will save the 'NHDFlowline_Network' and 'Catchment' or 'CatchmentSP' (per the 'simplified' parameter) as sf data.frames with superfluous Z information dropped.

The returned list of paths is also added to the nhdplusTools_env as "national_data".

Value

list containing paths to the .rds files.

Examples

```
sample_data <- system.file("extdata/sample_natseamless.gpkg",
                           package = "nhdplusTools")

stage_national_data(nhdplus_data = sample_data, output_path = tempdir())
```

subset_nhdplus *Subset NHDPlus*

Description

Saves a subset of the National Seamless database or other nhdplusTools compatible data based on a specified collection of COMIDs.

Usage

```
subset_nhdplus(comids, output_file, nhdplus_data = NULL,
  simplified = TRUE, overwrite = FALSE, status = TRUE)
```

Arguments

comids	integer vector of COMIDs to include.
output_file	character path to save the output to defaults to the directory of the nhdplus_data.
nhdplus_data	character path to the .gpkg or .gdb containing the national seamless database, a subset of NHDPlusHR, or "download" to use a web service to download NHDPlusV2.1 data. Not required if nhdplus_path has been set or the default has been adopted. See details for more.
simplified	boolean if TRUE (the default) the CatchmentSP layer will be included. Not relevant to the "download" option or NHDPlusHR data.
overwrite	boolean should the output file be overwritten
status	boolean should the function print status messages

Details

If stage_national_data has been run in the current session, this function will use the staged national data automatically.

This function relies on the National Seamless Geodatabase or Geopackage. It can be downloaded here.

The "download" option of this function should be considered preliminary and subject to revision. It does not include as many layers and may not be available permenantly.

Value

path to the saved subset geopackage

Examples

```
sample_data <- system.file("extdata/sample_natseamless.gpkg",
                           package = "nhdplusTools")

nhdplus_path(sample_data)
```

```
staged_nhdplus <- stage_national_data(output_path = tempdir())

sample_flines <- readRDS(staged_nhdplus$flowline)

geom_col <- attr(sample_flines, "sf_column")

plot(sample_flines[[geom_col]],
     lwd = 3)

start_point <- sf::st_sfc(sf::st_point(c(-89.362239, 43.090266)),
                          crs = 4326)

plot(start_point, cex = 1.5, lwd = 2, col = "red", add = TRUE)

start_comid <- discover_nhdplus_id(start_point)

comids <- get_UT(sample_flines, start_comid)

plot(dplyr::filter(sample_flines, COMID %in% comids)[[geom_col]],
     add=TRUE, col = "red", lwd = 2)

output_file <- tempfile(fileext = ".gpkg")

subset_nhdplus(comids = comids,
               output_file = output_file,
               nhdplus_data = sample_data,
               overwrite = TRUE,
               status = TRUE)

sf::st_layers(output_file)

catchment <- sf::read_sf(output_file, "CatchmentSP")

plot(catchment[[attr(catchment, "sf_column")]], add = TRUE)

waterbody <- sf::read_sf(output_file, "NHDWaterbody")

plot(waterbody[[attr(waterbody, "sf_column")]],
     col = rgb(0, 0, 1, alpha = 0.5), add = TRUE)

# Download Option:
subset_nhdplus(comids = comids,
               output_file = output_file,
               nhdplus_data = "download",
               overwrite = TRUE,
               status = TRUE)

sf::st_layers(output_file)

# NHDPlusHR
temp_dir <- tempdir()
temp_file <- tempfile(fileext = ".zip", tmpdir = temp_dir)
```

```
download.file("https://usgs-r.github.io/nhdplusTools/data/03_sub.zip",
              temp_file)
unzip(temp_file, exdir = temp_dir)

hr_data <- get_nhdplushr(temp_dir,
                         out_gpkg = file.path(temp_dir, "nhd_hr.gpkg"),
                         layers = NULL)
flowlines <- sf::read_sf(hr_data, "NHDFlowline")

up_ids <- get_UT(flowlines, 15000500028335)

sub_nhdhr <- subset_nhdplus(up_ids, file.path(temp_dir, "sub.gpkg"),
                      hr_data, overwrite = TRUE)

sf::st_layers(sub_nhdhr)

sub_flowline <- sf::read_sf(sub_nhdhr, "NHDFlowline")
plot(sf::st_geometry(flowlines), lwd = 0.5)
plot(sf::st_geometry(sub_flowline), lwd = 0.6, col = "red", add = TRUE)
```

Index