Package ‘nhdR’

July 17, 2021

Title Tools for Working with the National Hydrography Dataset

Version 0.5.6

Description Tools for working with the National Hydrography Dataset, with
functions for querying, downloading, and networking both the NHD

URL https://github.com/jsta/nhdR

BugReports https://github.com/jsta/nhdR/issues

Depends R (>= 3.5.0), maps

License GPL

Imports rappdirs, rgdal, sf, httr, rvest, xml2, foreign, ggplot2,
rlang, dplyr, curl, units, stringr, memoise, purrr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, wikilake, sp, rgeos, testthat (>= 2.1.0),
coverr, crul, lwgeom, s2

VignetteBuilder knitr

SystemRequirements 7-zip command line tool (7z)

Language en-US

NeedsCompilation no

Author Jemma Stachelek [aut, cre] (<https://orcid.org/0000-0002-5924-2464>)

Maintainer Jemma Stachelek <stachel2@msu.edu>

Repository CRAN

Date/Publication 2021-07-17 10:00:02 UTC
R topics documented:

- nhdR-package .................................................. 2
- bbox2poly .................................................... 3
- extract_network .............................................. 3
- find_vpu ....................................................... 5
- great_lakes .................................................... 6
- gull ............................................................ 6
- gull_flow ....................................................... 7
- leaf_reaches ................................................... 7
- mendota ........................................................ 8
- mendota_network ............................................. 8
- nhd_get ........................................................ 9
- nhd_info ....................................................... 9
- nhd_list ........................................................ 10
- nhd_load ....................................................... 11
- nhd_plus_get ................................................ 12
- nhd_plus_info ............................................... 13
- nhd_plus_list ............................................... 13
- nhd_plus_load ............................................... 14
- nhd_plus_query ............................................. 15
- nhd_query ..................................................... 17
- select_point_overlay .................................... 18
- select_poly_overlay .................................... 18
- sunapee ....................................................... 19
- sunapee_network .......................................... 19
- terminal_reaches ......................................... 20
- toUTM ........................................................ 21
- vpu_shp ....................................................... 22

Index 23

R interface to the National Hydrography Dataset

Description

R interface to the National Hydrography Dataset

Author(s)

<stachel2@msu.edu>
### bbox2poly

*Convert a bounding box to polygon*

**Description**

Convert a bounding box to polygon

**Usage**

```r
bbox2poly(bbox)
```

**Arguments**

- `bbox`: object of class bbox from sf

**Value**

An sfc object from the sf package

**Examples**

```r
## Not run:
library(sf)
wk <- wikilake::lake_wiki("Gull Lake (Michigan)")
pnt <- st_as_sf(wk, coords = c("Lon", "Lat"), crs = 4326)
pnt <- st_transform(pnt, st_crs(vpu_shp))
qry <- nhd_plus_query(wk$Lon, wk$Lat,
  dsn = c("NHDWaterbody"), buffer_dist = 0.05)
wbdf <- qry$sp$NHDWaterbody[which.max(st_area(qry$sp$NHDWaterbody)), ]
bbox2poly(st_bbox(wbdf))
## End(Not run)
```

---

### extract_network

*Return nhd plus stream network upstream of a waterbody*

**Description**

Return nhd plus stream network upstream of a waterbody
extract_network

Usage

extract_network(
  lon = NA,
  lat = NA,
  lines = NA,
  buffer_dist = 0.01,
  maxsteps = 3,
  approve_all_dl = FALSE,
  ...
)

Arguments

lon      numeric decimal degree longitude
lat      numeric decimal degree latitude
lines    sf spatial lines object to limit the extent of the network search
buffer_dist numeric buffer around lat-lon point in dec. deg.
maxsteps maximum number of stream climbing iterations
approve_all_dl logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
... parameters passed on to sf::st_read

Details

The lon and lat arguments are used for querying the corresponding lake polygon layer which is then used to climb its intersecting stream network.

Value

An sf data frame with LINESTRING geometries

Examples

## Not run:
library(mapview)
library(sf)

  # headwater lakes have no upstream network
  coords <- data.frame(lat = 46.32711, lon = -89.58893)
  res <- extract_network(coords$lon, coords$lat, maxsteps = 9)

  # fails if no lake nhdp lake found within the buffer at the query point
  coords <- data.frame(lat = 43.62453, lon = -85.47164)
  res <- extract_network(coords$lon, coords$lat, maxsteps = 9)

  coords <- data.frame(lat = 20.79722, lon = -156.47833)
  # use a non-geographic (projected) buffer size
  res <- extract_network(coords$lon, coords$lat, maxsteps = 9,
buffer_dist = units::as_units(5, "km")

# use a projected buffer size
res <- extract_network(coords$lon, coords$lat, maxsteps = 9)

# no upstream network for lakes intersecting the Great Lakes
coords <- data.frame(lat = 44.6265, lon = -86.23121)
res <- extract_network(coords$lon, coords$lat, maxsteps = 3)

coords <- data.frame(lat = 42.96523, lon = -89.2527)
res <- extract_network(coords$lon, coords$lat, maxsteps = 9)

mapview(res)

## End(Not run)

---

### Description

Find Vector Processing Unit from sf object

### Usage

```
find_vpu(pnt)
```

### Arguments

- **pnt**: sf object

### Value

A character vector of vpu ids

### Examples

```
## Not run:
library(sf)

# vpu centers
pnt <- st_cast(st_point_on_surface(nhdR::vpu_shp), "POINT")

find_vpu(pnt[1, ])
find_vpu(pnt)

find_vpu(nhdR::gull$sp$NHDWaterbody[1, ])
find_vpu(nhdR::gull$sp$NHDWaterbody)

## End(Not run)
```
great_lakes  

Data and spatial polygons of the Great Lakes

Description

Data and spatial polygons of the Great Lakes

Usage

great_lakes(spatial = FALSE)

Arguments

spatial logical, return Great Lakes polygons?

Value

A data frame of North America Great Lakes with optional geometry column

Examples

gl <- great_lakes()
## Not run:
gl <- great_lakes(spatial = TRUE)
## End(Not run)

---

gull  

List of simple features lake polygons and flowlines within a buffer around Gull Lake Michigan.

Description

Data from NHD Plus

Details

gull
**gull_flow**

*Flowlines within a buffer around Gull Lake Michigan including flow information.*

**Description**

Data from NHD Plus

**Details**

**gull_flow**

---

**leaf_reaches**

*Return leaf reaches from a network or query intersecting lake*

**Description**

A leaf reach is a stream flowline that has upstream connections but is not in the focal set.

**Usage**

```r
leaf_reaches(lon = NA, lat = NA, network = NA, approve_all_dl = FALSE, ...)
```

**Arguments**

- `lon` numeric decimal degree longitude. optional. See Details section.
- `lat` numeric decimal degree latitude. optional. See Details section.
- `network` sf lines collection. optional. See Details section.
- `approve_all_dl` logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
- `...` parameters passed on to sf::st_read

**Value**

An sf data frame with LINESTRING geometries
Examples

```r
## Not run:
coords <- data.frame(lat = 20.79722, lon = -156.47833)
leaf_reaches(coords$lon, coords$lat)

cords <- data.frame(lat = 41.42217, lon = -73.24189)
l_reach <- leaf_reaches(coords$lon, coords$lat)

network <- nhd_plus_query(lon = coords$lon, lat = coords$lat,
                           dsn = "NHDFlowline", buffer_dist = 0.02)$sp$NHDFlowline
l_reach <- leaf_reaches(network = network)

plot(network$geometry)
plot(l_reach$geometry, col = "red", add = TRUE)

## End(Not run)
```

---

mendota

List of simple features lake polygons and flowlines within a buffer around Lake Mendota.

---

Description

Data from NHD Plus

Details

mendota

---

mendota_network

Upstream flowlines connected to Lake Mendota.

---

Description

Data from NHD Plus

Details

mendota_network
**nhd_get**

*Download and cache NHD data by state*

---

**Description**

Download and cache NHD data by state

**Usage**

```r
nhd_get(state = NA, force_dl = FALSE, force_unzip = FALSE, temporary = TRUE)
```

**Arguments**

- **state** character state abbreviation includes "DC", "PR", and "VI"
- **force_dl** logical force a re-download of the requested data
- **force_unzip** logical force an unzip of downloaded data
- **temporary** logical set FALSE to save data to a persistent rappdirs location

**Value**

An invisible list of file paths to NHD data for the specified state

**Examples**

```r
## Not run:
nhd_get(state = c("DC"))
nhd_get(state = c("RI", "CT"))
## End(Not run)
```

---

**nhd_info**

*Return NHD layer metadata and field listing*

---

**Description**

Return NHD layer metadata and field listing

**Usage**

```r
nhd_info(state, dsn)
```

**Arguments**

- **state** character
- **dsn** character
### Description

List available locally cached NHD layers per state

### Usage

```r
dhc_list(state)
```

### Arguments

- `state` character state abbreviation

### Value

A character vector of NHD layers for the specified state

### Examples

```r
## Not run:
nhd_list(state = "DC")
## End(Not run)
```
**nhd_load**

Load NHD layers into current session

**Description**

Load NHD layers into current session

**Usage**

`nhd_load(state, dsn, file_ext = NA, approve_all_dl = FALSE, ...)`

**Arguments**

- **state**: character state abbreviation
- **dsn**: character name of a NHD layer
- **file_ext**: character choice of "shp" for spatial data and "dbf" or "gpkg" for non-spatial, optional
- **approve_all_dl**: logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
- **...**: arguments passed to `sf::st_read`

**Details**

This function will ask the user to approve downloading missing data unless `approve_all_dl` is set to TRUE.

**Value**

Spatial simple features object or data frame depending on the `dsn` type and value passed to `file_ext`

**Examples**

```r
## Not run:
dt <- nhd_load(c("RI"), c("NHDWaterbody"))
dt <- nhd_load(c("CT", "RI"), "NHDWaterbody")
dt <- nhd_load(c("CT", "RI"), "NHDWaterbody", quiet = TRUE)
dt <- nhd_load("MI", "NHDFlowline")
dt <- nhd_load("RI", "NHDReachCrossReference")
dt <- nhd_load("RI", "NHDWaterbody", file_ext = "dbf")
dt <- nhd_load(c("RI", "DC"), "NHDWaterbody", file_ext = "gpkg")
## End(Not run)
```
**nhd_plus_get**  

*Download and cache NHDplus data by vector processing unit*

**Description**

Download and cache NHDplus data by vector processing unit

**Usage**

```r
nhd_plus_get(
  vpu = NA,
  component = "NHDSnapshot",
  force_dl = FALSE,
  force_unzip = FALSE,
  temporary = TRUE
)
```

**Arguments**

- **vpu**: numeric vector processing unit
- **component**: character component name
- **force_dl**: logical force a re-download of the requested data
- **force_unzip**: logical force an unzip of downloaded data
- **temporary**: logical set FALSE to save data to a persistent rappdirs location

**Value**

An invisible list of file paths to NHDplus data for the specified vpu

**Examples**

```r
## Not run:  
# Spatial  
nhd_plus_get(vpu = 4)  
nhd_plus_get(vpu = "10L")  
nhd_plus_get(vpu = 1, component = "NHDPlusAttributes")

# Non-spatial  
nhd_plus_get(vpu = "National", component = "V1_To_V2_Crosswalk")  
nhd_plus_get(vpu = 4, component = "EROMExtension")

## End(Not run)
```
nhd_plus_info

Return NHDplus layer metadata and field listing

Description
Return NHDplus layer metadata and field listing

Usage
nhd_plus_info(vpu, component, dsn, file_ext = NA)

Arguments
vpu numeric vector processing unit
component character component name
dsn character data source name
file_ext character choice of "shp" for spatial data and "dbf" for non-spatial. optional

Value
An ogrinfo object from the sf package

Examples
## Not run:
nhd_plus_info(vpu = 4, component = "NHDSnapshot", dsn = "NHDWaterbody")
nhd_plus_info(vpu = 1, component = "NHDPlusAttributes", dsn = "PlusFlow")

## End(Not run)

nhd_plus_list
List available locally cached NHDplus layers per state

Description
List available locally cached NHDplus layers per state

Usage
nhd_plus_list(vpu, component = "NHDSnapshot", file_ext = NA, ...)

Arguments
vpu numeric vector processing unit
component character component name
file_ext character choice of "shp" for spatial data and "dbf" for non-spatial. optional
... arguments passed to list.files. optional.
Value

A character vector of NHD layers for the specified vpu

Examples

```r
## Not run:
nhd_plus_list(vpu = 4)
nhd_plus_list(vpu = 4, full.names = TRUE)

nhd_plus_list(vpu = 1, component = "NHDPlusAttributes")
nhd_plus_list(vpu = "National", component = "V1_To_V2_Crosswalk")

## End(Not run)
```

---

**nhd_plus_load**

Load NHDplus layers into current session

### Description

Load NHDplus layers into current session

### Usage

```r
nhd_plus_load(
  vpu,
  component = "NHDSnapshot",
  dsn,
  file_ext = NA,
  approve_all_dl = FALSE,
  force_dl = FALSE,
  pretty = FALSE,
  wkt_filter = NA,
  ...
)
```

### Arguments

- **vpu**: numeric vector processing unit
- **component**: character component name
- **dsn**: data source name
- **file_ext**: character choice of "shp" for spatial data and "dbf" for non-spatial. optional
- **approve_all_dl**: logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive
- **force_dl**: logical force a re-download of the requested data
- **pretty**: more minimal pretty printing st_read relative to "quiet"
- **wkt_filter**: character. WKT spatial filter for selection. See sf::st_read
- **...**: parameters passed on to sf::st_read
Details

This function will ask the user to approve downloading missing data unless approve_all_dl is set to TRUE. Output of this function is saved in active memory (memoized) to speed up repeated function calls.

Value

spatial object

Examples

```r
## Not run:
# Spatial
dt <- nhd_plus_load(4, "NHDSnapshot", "NHDWaterbody")
dt <- nhd_plus_load(c(1,2), "NHDSnapshot", "NHDWaterbody")
dt <- nhd_plus_load(4, "NHDSnapshot", "NHDFlowline")
dt <- nhd_plus_load(4, "NHDPlusCatchment", "Catchment")

# Quieter printing
dt <- nhd_plus_load(4, "NHDSnapshot", "NHDWaterbody", pretty = TRUE)
# Quietest printing
dt <- nhd_plus_load(4, "NHDSnapshot", "NHDWaterbody", quiet = TRUE)

# Non-spatial
dt <- nhd_plus_load(1, "NHDPlusAttributes", "PlusFlow")
dt <- nhd_plus_load("National", "V1_To_V2_Crosswalk", "NHDPlusV1Network_V2Network_Crosswalk")
gridcode <- nhd_plus_load(1, "NHDPlusCatchment", "featuregridcode")
flowline_vaa <- nhd_plus_load(1, "NHDPlusAttributes", "PlusFlowlineVAA")
eromflow <- nhd_plus_load(4, "EROMExtension", "EROM_010001")

# Character VPU
plusflow <- nhd_plus_load(vpu = "10L", "NHDPlusAttributes", "PlusFlow")

# Spatial filtering via wkt_filter
dt <- nhd_plus_load(4, "NHDSnapshot", "NHDWaterbody", wkt_filter = "POINT (-85.411 42.399)")
```

## End(Not run)

---

**nhd_plus_query**

Select **NHDplus features via polygon or circular buffer of coordinate pair**

**Description**

Select NHDplus features via polygon or circular buffer of coordinate pair
Usage

nhd_plus_query(
  lon = NA,
  lat = NA,
  poly = NA,
  dsn,
  buffer_dist = 0.05,
  approve_all_dl = FALSE,
  ...
)

Arguments

- **lon**: numeric longitude. optional
- **lat**: numeric latitude. optional
- **poly**: sfc polygon. optional
- **dsn**: character data source
- **buffer_dist**: numeric buffer in units of coordinate degrees
- **approve_all_dl**: logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
- **...**: parameters passed on to sf::st_read

Value

A list of sf spatial objects

Examples

```r
## Not run:
library(sf)
wk <- wikilake::lake_wiki("Gull Lake (Michigan)")

pnt <- st_as_sf(wk, coords = c("Lon", "Lat"), crs = 4326)
pnt <- st_transform(pnt, st_crs(vpu_shp))
# nhd_plus_list(nhdR::find_vpu(pnt))

# set a non-geographic (projected) buffer size
qry <- nhd_plus_query(wk$Lon, wk$Lat,
  dsn = c("NHDWaterbody", "NHDFlowLine"),
  buffer_dist = units::as_units(5, "km"))

qry <- nhd_plus_query(wk$Lon, wk$Lat,
  dsn = c("NHDWaterbody", "NHDFlowLine"), buffer_dist = 0.05)

plot(qry$sp$NHDWaterbody$geometry, col = "blue")
plot(qry$sp$NHDFlowLine$geometry, col = "cyan", add = TRUE)
plot(qry$pnt, col = "red", pch = 19, add = TRUE)
axis(1)
```
nhd_query

Select NHD features clipped by a circular buffer a coordinate pair

Description

Select NHD features clipped by a circular buffer a coordinate pair

Usage

nhd_query(lon, lat, dsn, buffer_dist = 0.05)

Arguments

lon numeric longitude
lat numeric latitude
dsn character data source
buffer_dist numeric buffer in units of coordinate degrees

Examples

## Not run:
wk <- wikilake::lake_wiki("Worden Pond")
qry <- nhd_query(wk$lon, wk$lat, dsn = c("NHDWaterbody", "NHDFlowline"))

plot(sf::st_geometry(qry$sp$NHDWaterbody), col = "blue")
plot(sf::st_geometry(qry$sp$NHDFlowline), col = "cyan", add = TRUE)
plot(qry$pnt, col = "red", pch = 19, add = TRUE)
axis(1)
axis(2)

## End(Not run)
select_point_overlay  
Select features clipped by a point buffer around a point

Description
Select features clipped by a point buffer around a point

Usage
select_point_overlay(pnt, sp, buffer_dist = 0.05)

Arguments
- pnt: geographic point of class sfc
- sp: list of sf data frames
- buffer_dist: numeric buffer in units of coordinate degrees

Value
A list of sf spatial objects

Examples
```r
## Not run:
wk <- wikilake::lake_wiki("Gull Lake (Michigan)")
pnt <- sf::st_sfc(sf::st_point(c(wk$Lon, wk$Lat)))
sf::st_crs(pnt) <- 4326
sp <- lapply(c("NHDWaterbody", "NHDFlowLine"),
function(x) nhd_plus_load(vpu = 4, dsn = x))
names(sp) <- c("NHDWaterbody", "NHDFlowLine")
qry <- select_point_overlay(pnt = pnt, sp = sp, buffer_dist = 0.05)
plot(qry$NHDWaterbody$geometry)
## End(Not run)
```

select_poly_overlay  
Select features clipped by a polygon

Description
Select features clipped by a polygon

Usage
select_poly_overlay(poly, sp)
Arguments

poly sf *polygon object
sp list of sf data frames

Value

A list of sf spatial objects

---

sunapee

List of simple features lake polygons and flowlines within a buffer around Lake Sunapee.

---

Description

Data from NHD Plus

Details

sunapee

---

sunapee_network

Upstream flowlines connected to Lake Sunapee.

---

Description

Data from NHD Plus

Details

sunapee_network
terminal_reaches  

Return terminal reaches from collection intersecting lake

Description

In the case of a network query, a terminal reach is a stream flowline that has no downstream reaches in-network. In the case of a point query, a terminal reach is a flowline that exits the intersecting surface waterbody.

Usage

```r
terminal_reaches(
  lon = NA,
  lat = NA,
  buffer_dist = 0.01,
  network = NA,
  lakepoly = NA,
  lakewise = FALSE,
  lakesize_threshold = 4,
  approve_all_dl = FALSE,
  ...
)
```

Arguments

- `lon` numeric decimal degree longitude. optional. See Details section.
- `lat` numeric decimal degree latitude. optional. See Details section.
- `buffer_dist` numeric buffer around lat-lon point in dec. deg.
- `network` sf lines collection. optional. See Details section.
- `lakepoly` sf polygon. optional. See Details section.
- `lakewise` logical. If TRUE, return the terminal reaches of all lakes. in the stream network rather than a single terminal reach of the focal lake.
- `lakesize_threshold` numeric above which to count as a lake (ha).
- `approve_all_dl` logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
- `...` parameters passed on to sf::st_read

Details

There are multiple ways to execute `terminal_reaches`:

- Only providing lon + lat arguments - this will query the corresponding lake polygon layer and find the terminal reach of the lake intersecting a buffer around the specified point.
- Only providing a lake polygon - this is essentially the same as above except there is no preliminary lake polygon query.
- Only providing a network of stream lines - this provides the most downstream reach irrespective of lakes.

### Value

An sf data frame with LINESTRING geometries

### Examples

```r
## Not run:
library(sf)
library(mapview)

coords <- data.frame(lat = 46.32711, lon = -89.58893)
t_reach <- terminal_reaches(coords$lon, coords$lat)

coords <- data.frame(lat = 20.79722, lon = -156.47833)
# use a non-geographic (projected) buffer size
  t_reach <- terminal_reaches(coords$lon, coords$lat,
                          buffer_dist = units::as_units(5, "km"))

coords <- data.frame(lat = 42.96628, lon = -89.25264)
t_reach <- terminal_reaches(coords$lon, coords$lat)

coords <- data.frame(lat = 41.42217, lon = -73.24189)
t_reach <- terminal_reaches(coords$lon, coords$lat)

mapview(st_as_sf(coords, coords = c("lon", "lat"), crs = 4326)) +
  mapview(t_reach$geometry, color = "red")

coords <- data.frame(lat = 41.859080, lon = -71.575422)
network <- nhd_plus_query(lon = coords$lon, lat = coords$lat,
                          dsn = "NHDFlowline", buffer_dist = 0.05)$sp$NHDFlowline
  t_reach <- terminal_reaches(network = network)
t_reach_lake <- terminal_reaches(network = network, lakewise = TRUE,
                               lakesize_threshold = 1)

mapview(network) + mapview(t_reach_lake, color = "green") +
  mapview(t_reach, color = "red")

## End(Not run)
```

<table>
<thead>
<tr>
<th>toUTM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Re-project to appropriate UTM zone</strong></td>
</tr>
</tbody>
</table>

### Description

Re-project to appropriate UTM zone
Usage

toUTM(sf_object)

Arguments

sf_object an sf object

Value

A transformed sf object

Examples

## Not run:
data(gull)
gull_ <- gull$sp$NHDWaterbody
st_crs(gull_)
gull_ <- st_transform(gull_, 4326)
st_crs(gull_)
st_crs(toUTM(gull_[1, ]))

## End(Not run)

vpu_shp Low-res simple features data frame of the NHDPlus vector processing units

Description

vpu_shp
Index

* datasets
  gull, 6
  gull_flow, 7
  mendota, 8
  mendota_network, 8
  sunapee, 19
  sunapee_network, 19
  vpu_shp, 22

bbox2poly, 3
extract_network, 3
find_vpu, 5
great_lakes, 6
gull, 6
gull_flow, 7
leaf_reaches, 7
mendota, 8
mendota_network, 8

nhd_get, 9
nhd_info, 9
nhd_list, 10
nhd_load, 11
nhd_plus_get, 12
nhd_plus_info, 13
nhd_plus_list, 13
nhd_plus_load, 14
nhd_plus_query, 15
nhd_query, 17
nhdR (nhdR-package), 2
nhdR-package, 2

select_point_overlay, 18
select_poly_overlay, 18
sunapee, 19
sunapee_network, 19

terminal_reaches, 20, 20
toUTM, 21

vpu_shp, 22