Package 'dbhydroR'

May 8, 2026

Title 'DBHYDRO' Hydrologic and Water Quality Data

Description Client for programmatic access to the South Florida Water Management District's 'DBHYDRO' database at <https://www.sfwmd.gov/science-data/dbhydro>, with functions for accessing hydrologic and water quality data.

Version 0.2-8

URL https://github.com/ropensci/dbhydroR, https://docs.ropensci.org/dbhydroR/

BugReports https://github.com/ropensci/dbhydroR/issues

Depends R (>= 3.0.2)

Imports httr, reshape2, XML

License GPL

LazyData true

Suggests testthat, knitr, rmarkdown, vcr

RoxygenNote 7.1.1

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

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

Maintainer Joseph Stachelek <stachel2@msu.edu>

Repository CRAN

Date/Publication 2021-02-21 16:50:02 UTC

Contents

`dbhydroR-package` *dbhydroR*

Description

dbhydroR is an R interface to the South Florida Water Management District's DBHYDRO database which holds over 35 million hydrologic and water quality records from the Florida Everglades and surrounding areas.

Hydrologic Data

The `get_hydro` function provides the capability to return:

- weather data
- surfacewater data
- groundwater data
- water-quality sonde data

Water Quality Data

The `get_wq` function provides the capability to return:

- water quality data

`clean_hydro` *Clean raw hydrologic DBHYDRO data retrievals*

Description

Converts output of `get_hydro` from long (each piece of data on its own row) to wide format (each site x variable combination in its own column). Metadata (station-name, variable, measurement units) is parsed so that it is wholly contained in column names.

Usage

```
clean_hydro(dt)
```

Arguments

`dt`	data.frame output of `gethydro`

Examples

```
## Not run:
clean_hydro(gethydro(dbkey = "15081", date_min = "2013-01-01", date_max = "2013-02-02", raw = TRUE))

## End(Not run)
```

---

clean_wq *Clean raw water quality DBHYDRO data retrievals*

Description

Removes extra columns associated with QA flags and QA blanks which are used to check on potential sources of contamination. If raw is set to TRUE, get_wq results are converted from long (each piece of data on its own row) to wide format (each site x variable combination in its own column).

Usage

```
clean_wq(dt, raw = FALSE, mdl_handling = "raw")
```

Arguments

dt	data.frame output of getwq
raw	logical default is FALSE, set to TRUE to return data in "long" format with all comments, qa information, and database codes included
mdl_handling	character string specifying the handling of measurement values below the minimum detection limit (MDL). Example choices for this argument include: • raw: Returns values exactly as they are stored in the database. Current practice is to return values below the MDL as 0 minus the uncertainty estimate. • half: Returns values below the MDL as half the MDL • full: Returns values below the MDL as the MDL

Examples

```
## Not run:
#check handling of values below MDL
dt <- getwq("FLAB01", "2014-09-14", "2014-09-18", "NITRATE+NITRITE-N",
 raw = TRUE)
clean_wq(dt, mdl_handling = "raw")
clean_wq(dt, mdl_handling = "half")

## End(Not run)

dt <- read.csv(system.file("extdata", "testwq.csv", package = "dbhydroR"))
clean_wq(dt)
```

get_dbkey *Query dbkey information*

Description

Retrieve a data.frame summary including dbkeys or a vector of dbkeys corresponding to specified parameters

Usage

```
get_dbkey(
  category,
  stationid = NA,
  param = NA,
  freq = NA,
  longest = FALSE,
  stat = NA,
  recorder = NA,
  agency = NA,
  strata = NA,
  detail.level = "summary",
  ...
)
```

Arguments

category	character string, choice of "WEATHER", "SW", "GW", or "WQ"
stationid	character string specifying station name
param	character string specifying desired parameter name
freq	character string specifying collection frequency (daily = "DA")
longest	logical limit results to the longest period-of-record?
stat	character string specifying statistic type
recorder	character string specifying recorder information
agency	character string specifying collector agency
strata	numeric vector of length 2 specifying a range of z-coordinates relative to local ground elevation. Only applicable for queries in the "WEATHER" and "GW" categories.
detail.level	character string specifying the level of detail to return. Choices are "full", "summary", and "dbkey".
...	Options passed as named parameters

Details

A dbkey represents a unique station x variable time-series. A value in the "Recorder" field of "PREF" should be used whenever possible. This indicates that the dataset has been checked by the SFWMD modelling group.

Examples

```
## Not run:
# Weather
get_dbkey(stationid = "JBTS", category = "WEATHER", param = "WNDS",
detail.level = "summary")
get_dbkey(stationid = "JBTS", category = "WEATHER", param = "WNDS",
detail.level = "dbkey")

# query on multiple values
get_dbkey(stationid = c("MBTS", "JBTS"), category = "WEATHER",
param = "WNDS", freq = "DA", detail.level = "dbkey")


# Surfacewater
get_dbkey(stationid = "C111%", category = "SW")
get_dbkey(category = "SW", stationid = "LAKE%", detail.level = "full")

# Groundwater
get_dbkey(stationid = "C111%", category = "GW")
get_dbkey(stationid = "C111AE", category = "GW", param = "WELL",
freq = "DA", stat = "MEAN", strata = c(9, 22), recorder = "TROL",
 agency = "WMD", detail.level = "full")

# Water Quality
get_dbkey(stationid = "C111%", category = "WQ")

## End(Not run)
```

`get_hydro` *Retrieve hydrologic data from the DBHYDRO Environmental Database*

Description

Retrieve hydrologic data from the DBHYDRO Environmental Database

Usage

```
get_hydro(dbkey = NA, date_min = NA, date_max = NA, raw = FALSE, ...)
```

Arguments

`dbkey`	character string specifying a unique data series. See `get_dbkey`
`date_min`	character date must be in YYYY-MM-DD format
`date_max`	character date must be in YYYY-MM-DD format
`raw`	logical default is FALSE, set to TRUE to return data in "long" format with all comments, qa information, and database codes included.
`...`	Options passed on to `get_dbkey`

Details

get_hydro can be run in one of two ways.

- The first, is to identify one or more dbkeys before-hand that correspond to unique data series and are passed to the dbkey argument. dbkeys can be found by:
  - iterative calls to get_dbkey (see example)
  - using the Environmental Monitoring Location Maps (https://www.sfwmd.gov/documents-by-tag/emmaps)
  - using the DBHYDRO Browser.
- The second way to run get_hydro is to specify additional arguments to ... which are passed to get_dbkey on-the-fly.

By default, get_hydro returns a cleaned output where metadata (station-name, variable, measurement units) is wholly contained in the column name. This is accomplished internally by the clean_hydro function. If additional metadata such as latitude and longitude are desired set the raw argument to TRUE.

Examples

```
## Not run:
#One variable/station time series
get_hydro(dbkey = "15081", date_min = "2013-01-01", date_max = "2013-02-02")

#Multiple variable/station time series
get_hydro(dbkey = c("15081", "15069"),
date_min = "2013-01-01", date_max = "2013-02-02")

#Instantaneous hydro retrieval
get_hydro(dbkey = "IY639", date_min = "2015-11-01", date_max = "2015-11-04")

#Looking up unknown dbkeys on the fly
get_hydro(stationid = "JBTS", category = "WEATHER",
param = "WNDS", freq = "DA", date_min = "2013-01-01",
date_max = "2013-02-02")

## End(Not run)
```

---

get_wq *Retrieve water quality data from the DBHYDRO Environmental Database*

Description

Retrieve water quality data from the DBHYDRO Environmental Database

Usage

```
get_wq(
  station_id = NA,
  date_min = NA,
  date_max = NA,
  test_name = NA,
  mdl_handling = "raw",
  raw = FALSE,
  qc_strip = "N",
  qc_field = "N",
  test_number = NA,
  v_target_code = "file_csv",
  sample_id = NA,
  project_code = NA
)
```

Arguments

station_id	character string of station id(s). See the SFWMD station search utility for specific options
date_min	character date must be in POSIXct YYYY-MM-DD format
date_max	character date must be in POSIXct YYYY-MM-DD format
test_name	character string of test name(s). See the SFWMD Station Maps at https://www.sfwmd.gov/documents-by-tag/emmaps for specific options
mdl_handling	character string specifying the handling of measurement values below the minimum detection limit (MDL). Example choices for this argument include: • raw: Returns values exactly as they are stored in the database. Current practice is to return values below the MDL as 0 minus the uncertainty estimate. • half: Returns values below the MDL as half the MDL • full: Returns values below the MDL as the MDL
raw	logical default is FALSE, set to TRUE to return data in "long" format with all comments, qa information, and database codes included
qc_strip	logical set TRUE to avoid returning QAQC flagged data entries
qc_field	logical set TRUE to avoid returning field QC results
test_number	numeric test name alternative (not implemented)
v_target_code	string print to file? (not implemented)
sample_id	numeric (not implemented)
project_code	numeric (not implemented)

Details

By default, get_wq returns a cleaned output. First, the cleaning function clean_wq converts the raw output from native DBHYDRO long format (each piece of data on its own row) to wide format (each site x variable combination in its own column). Next, the extra columns associated with QA flags,

LIMS, and District receiving are removed. Finally, row entries associated with QA field blanks, which are used to check on potential sources of contamination, are removed. Setting the raw flag to TRUE will force getwq to retain information on QA field blanks as well as the other QA fields.

Examples

```
## Not run:
#one variable and one station
get_wq(station_id = "FLAB08",
date_min = "2011-03-01", date_max = "2012-05-01",
test_name = "CHLOROPHYLLA-SALINE")

#one variable at multiple stations
get_wq(station_id = c("FLAB08", "FLAB09"),
date_min = "2011-03-01", date_max = "2012-05-01",
test_name = "CHLOROPHYLLA-SALINE")

#One variable at a wildcard station
get_wq(station_id = c("FLAB0%"),
date_min = "2011-03-01",
date_max = "2012-05-01",
test_name = "CHLOROPHYLLA-SALINE")

#multiple variables at multiple stations
get_wq(station_id = c("FLAB08", "FLAB09"),
date_min = "2011-03-01", date_max = "2012-05-01",
test_name = c("CHLOROPHYLLA-SALINE", "SALINITY"))

## End(Not run)
```

Index