Package ‘rerddapXtracto’

June 2, 2021

Type Package
Title Extracts Environmental Data from 'ERDDAP' Web Services
Version 1.1.1
Date 2021-06-01
Description Contains three functions that access environmental data from any 'ERDDAP' data web service. The rxtracto() function extracts data along a trajectory for a given "radius" around the point. The rxtracto_3D() function extracts data in a box. The rxtractogon() function extracts data in a polygon. All of those three function use the 'rerddap' package to extract the data, and should work with any 'ERDDAP' server. There are also two functions, plotBBox() and plotTrack() that use the 'plotdap' package to simplify the creation of maps of the data.

URL https://github.com/rmendels/rerddapXtracto

BugReports https://github.com/rmendels/rerddapXtracto/issues
Depends R(>= 4.0.0)
License CC0
Imports abind, dplyr, ggplot2, htrr, maps, methods, ncdf4, parsedate, plotdap (>= 0.0.5), readr, rerddap (>= 0.6.0), sf, sp, stats, utils
Suggests cmocean, gganimate, knitr, mapdata, maptools, rgdal, rmarkdown
RoxygenNote 7.1.1
Encoding UTF-8
LazyData TRUE
VignetteBuilder knitr
NeedsCompilation no
Author Roy Mendelssohn [aut, cre]
Maintainer Roy Mendelssohn <roy.mendelssohn@noaa.gov>
Repository CRAN
Date/Publication 2021-06-02 07:20:02 UTC
R topics documented:

**dataInfo** .......................... 2
Marlintag38606 ........................ 2
mbnms .................................. 3
MBsst .................................. 4
plotBBBox .............................. 4
plotTrack .............................. 5
rerddapXtracto ........................ 7
rxtracto .............................. 7
rxtractogon ........................... 10
rxtracto_3D ........................... 11
swchl ................................. 13

Index .............................. 14

---

**dataInfo**

**dataInfo Data**

**Description**

pre-Download of `rerddap` info needed for examples so can run within CRAN Time limits

**Usage**

dataInfo

**Format**

An object of class `info` of length 3.

**Details**

obtained using `dataInfo <- rerddap::info('erdHadISST')`

---

**Marlintag38606**

**Marlin Tag Data**

**Description**

Telemetry data of a blue marlin tagged in the Pacific Ocean in 2003

**Usage**

Marlintag38606
Format

A data frame with 152 obs. of 7 variables:

- **date**  time of observation given as YYYY-MM-DD
- **lon**  longitude of observation
- **lat**  latitude of observation
- **lowLon**  low error on longitude
- **higLon**  high error on longitude
- **lowLat**  low error on latitude
- **higLat**  high error on latitude

Source

Dr. Mike Musyl, Pelagic Research Group LLC

---

**mbnms**  |  **MBNMS Boundaries**

Description

A dataset containing the latitudes and longitudes of a polygon that define boundaries of the Monterey Bay National Marine Sanctuary.

Usage

mbnms

Format

A data frame with 6666 obs. of 2 variables:

- **Longitude**  Longitudes of a boundary polygon
- **Latitude**  Latitudes of a boundary polygon

Source

https://sanctuaries.noaa.gov/library/imast_gis.html


### MBsst

**MBsst Data**

**Description**

pre-Download of Pacific West Coast SST for use in `plotBBox()` example can run within CRAN

**Time limits**

**Usage**

`MBsst`

**Format**

An object of class `list` (inherits from `rxtracto3D`) of length 6.

**Details**

obtained using the `rxtracto_3D()` command:

```r
dataInfo <- rerddap::info('erdMBsststd1day')
parameter <- 'sst'
xcoord <- c(230, 231)
ycoord <- c(33, 34)
tcoord <- c('2006-01-15', '2006-01-15')
zcoord <- c(0., 0.)
MBsst <- rxtracto_3D(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord,
tcoord = tcoord, zcoord = zcoord)
```

### plotBBox

**plot result of `rxtracto_3D`**

**Description**

`plotBBox` is a function to plot the results from `rxtracto()` and `xtracto()`

**Usage**

```r
plotBBox(
  resp,
  plotColor = "viridis",
  time = NA,
  myFunc = NA,
  mapData = NULL,
  crs = NULL,
  animate = FALSE,
  cumulative = FALSE,
  name = NA,
  maxpixels = 10000
)
```
**plotTrack**

**Arguments**

- **resp**: data frame returned from `rxtracto()`
- **plotColor**: the color to use in plot from `cmocean`
- **time**: a function to map multi-time to one, or else identity for animation
- **myFunc**: function of one argument to transform the data
- **mapData**: map data from `maps` or `mapdata`, must be of class `map`
- **crs**: valid crs string
- **animate**: if multiple times, if TRUE will animate the maps
- **cumulative**: makes cumulative animation of data
- **name**: name for colorbar label
- **maxpixels**: maximum number of pixels to use in making the map - controls resolution

**Value**

- a `plotdap` plot

**Examples**

```r
dataInfo <- rerddap::info('erdMBsst1day')
parameter <- 'sst'
xcoord <- c(230, 232)
ycoord <- c(33, 35)
tcoord <- c('2006-01-15', '2006-01-15')
zcoord <- c(0., 0.)
MBsst <- rxtracto_3D(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord,
tcoord = tcoord, zcoord = zcoord)

# low resolution selected to keep time to render down
suppressWarnings(p <- plotBBox(MBsst, plotColor = 'thermal', maxpixels = 300))
```

**Description**

`plotTrack` is a function to plot the results from `rxtracto()`

**Usage**

```r
plotTrack(
  resp,
  xcoord,
  ycoord,
  tcoord,
```
plotColor = "viridis",
myFunc = NA,
mapData = NULL,
crs = NULL,
animate = FALSE,
cumulative = FALSE,
name = NA,
shape = 20,
size = 0.5
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resp</td>
<td>data frame returned from 'rxtracto()'</td>
</tr>
<tr>
<td>xcoord</td>
<td>passed to 'rxtracto()'</td>
</tr>
<tr>
<td>ycoord</td>
<td>passed to 'rxtracto()'</td>
</tr>
<tr>
<td>tcoord</td>
<td>passed to 'rxtracto()'</td>
</tr>
<tr>
<td>plotColor</td>
<td>the color to use in plot from 'cmocean'</td>
</tr>
<tr>
<td>myFunc</td>
<td>function of one argument to transform the data</td>
</tr>
<tr>
<td>mapData</td>
<td>map data from 'maps' or 'mapdata', must be of class 'map'</td>
</tr>
<tr>
<td>crs</td>
<td>valid crs string</td>
</tr>
<tr>
<td>animate</td>
<td>if multiple times, if TRUE will animate the maps</td>
</tr>
<tr>
<td>cumulative</td>
<td>makes cumulative animation of data</td>
</tr>
<tr>
<td>name</td>
<td>name for colorbar label</td>
</tr>
<tr>
<td>shape</td>
<td>shape to use to mark track</td>
</tr>
<tr>
<td>size</td>
<td>size of shape to use to mark track</td>
</tr>
</tbody>
</table>

Value

a 'plotdap' plot

Examples

tagData <- Marlintag38606
xpos <- tagData$lon[1:20]
ypos <- tagData$lat[1:20]
tpos <- tagData$date[1:20]
zpos <- rep(0., length(xpos))

swchlInfo <- rerddap::info('erdSWchla8day')
swchl <- rxtracto(swchlInfo, parameter = 'chlorophyll', xcoord = xpos,
ycoord = ypos, tcoord = tpos, zcoord = zpos, xlen = .2, ylen = .2)
suppressWarnings(p <- plotTrack(swchl, xpos, ypos, tpos, plotColor = 'algae'))
rerddapXtracto

rerddapXtracto: Routines to simplify data extraction using ERD’s ERDDAP web service.

Description

The rerddapXtracto package subsets and extracts satellite and other oceanographic related data from any ERDDAP server using the R package "rerddap" developed by Scott Chamberlain and the wonderful people at "rOpenSci".

Details

The package contains three main functions:

Main Functions

- `rxtracto` - Extracts an environmental variable along a track defined by its longitude, latitude and time.
- `rxtracto_3D` - Extracts an environmental variable in a 3D (longitude, latitude, time) bounding box
- `rxtractogon` - Extracts an environmental variable in a spatial polygon through time.

These functions require a call be made to `rerddap::info()` for the appropriate ERDDAP server and datasetID. numeric. Besides the terse help documents, more detail in using the functions are given in the included vignette "UsingrerddapXtracto". The datasets used in the vignette are included in the "data" directory.

rxtracto

Extract environmental data along a trajectory from an 'ERDDAP' server using 'rerddap'.

Description

`rxtracto_new` uses the R program 'rerddap' to extract environmental data from an 'ERDDAP' server along a (x,y,z, time) trajectory.

Usage

```r
rxtracto(
  dataInfo,
  parameter = NULL,
  xcoord = NULL,
  ycoord = NULL,
  zcoord = NULL,
  tcoord = NULL,
  xlen = 0,
)```
ylen = 0,
zlen = 0,
xName = "longitude",
yName = "latitude",
zName = "altitude",
tName = "time",
interp = NULL,
verbose = FALSE,
progress_bar = FALSE
)

Arguments

dataInfo - the return from an 'rerddap::info' call to an 'ERDDAP' server
parameter - character string containing the name of the parameter to extract
xcoord - a real array with the x-coordinates of the trajectory (if longitude in #' decimal
degrees East, either 0-360 or -180 to 180)
ycoord - a real array with the y-coordinate of the trajectory (if latitude in decimal de-
grees N; -90 to 90)
zcoord - a real array with the z-coordinate of the trajectory (usually altitude or depth)
tcoord - a character array with the times of the trajectory in "YYYY-MM-DD" - for
now restricted to be time.
xlen - real array defining the longitude box around the given point (xlen/2 around the
point)
ylen - real array defining the latitude box around the given point (ylen/2 around the
point)
zlen - real array defining the depth or altitude box around the given point (zlen/2
around the point)
xName - character string with name of the xcoord in the 'ERDDAP' dataset (default
"longitude")
yName - character string with name of the ycoord in the 'ERDDAP' dataset (default
"latitude")
zName - character string with name of the zcoord in the 'ERDDAP' dataset (default
"altitude")
tName - character string with name of the tcoord in the 'ERDDAP' dataset (default
"time")
interp - array (size 2) of character strings - c(interpolation type, neighborhood) Uses
the new ERDDAP interpoation option to get values See Vignette for details De-
fault is Null, do not use the interpolation option
verbose - logical variable (default FALSE) if the the URL request should be verbose
progress_bar - logical variable (default FALSE) should a progress bar be displayed
Value

A dataframe containing:

- column 1 = mean of data within search radius
- column 2 = standard deviation of data within search radius
- column 3 = number of points found within search radius
- column 4 = time of returned value
- column 5 = min longitude of call (decimal degrees)
- column 6 = max longitude of call (decimal degrees)
- column 7 = min latitude of call (decimal degrees)
- column 8 = max latitude of call (decimal degrees)
- column 9 = requested time in tag
- column 10 = median of data within search radius
- column 11 = median absolute deviation of data within search radius

Examples

# toy example to show use
# but keep execution time down
dataInfo <- rerddap::info('erdHadISST')

parameter <- 'sst'
xcoord <- c(-130.5)
ycoord <- c(40.5)
tcoord <- c('2006-01-16')
xlen <- 0.01
ylen <- 0.01
extract <- rxtracto(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord, tcoord = tcoord, xlen = xlen, ylen = ylen)

# 2-D example getting bathymetry
dataInfo <- rerddap::info('etopo360')
parameter <- 'altitude'
extract <- rxtracto(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord, xlen = xlen, ylen = ylen)
**rxtractogon**

*Extract environmental data in a polygon using ‘ERDDAP’ and ‘rerddap’.*

### Description

*rxtractogon* uses the R program ‘rerddap’ to extract environmental data from an ‘ERDDAP’ server in a polygon through time.

### Usage

```r
rxtractogon(
  dataInfo,
  parameter,
  xcoord = NULL,
  ycoord = NULL,
  zcoord = NULL,
  tcoord = NULL,
  xName = "longitude",
  yName = "latitude",
  zName = "altitude",
  tName = "time",
  verbose = FALSE,
  cache_remove = TRUE
)
```

### Arguments

- **dataInfo** - the return from an ‘rerddap:info’ call to an ‘ERDDAP’ server
- **parameter** - character string containing the name of the parameter to extract
- **xcoord** - array giving longitudes (in decimal degrees East, either 0-360 or -180 to 180) of polygon
- **ycoord** - array giving latitudes (in decimal degrees N; -90 to 90) of polygon
- **zcoord** - a real number with the z-coordinate (usually altitude or depth)
- **tcoord** - 2-array of minimum and maximum times as ‘YYYY-MM-DD’
- **xName** - character string with name of the xcoord in the ‘ERDDAP’ dataset (default "longitude")
- **yName** - character string with name of the ycoord in the ‘ERDDAP’ dataset (default "latitude")
- **zName** - character string with name of the zcoord in the ‘ERDDAP’ dataset (default "altitude")
- **tName** - character string with name of the tcoord in the ‘ERDDAP’ dataset (default "time")
- **verbose** - logical variable (default FALSE) if the URL request should be verbose
- **cache_remove** - logical variable (default TRUE) whether to delete ‘rerddap’ cache
**rxtracto_3D**

*Extract environmental data in a 3-dimensional box from an 'ERDDAP' server using 'rerddap'.*

**Value**

structure with data and dimensions

- extract$data - the masked data array dimensioned (lon, lat, time)
- extract$svarname - the name of the parameter extracted
- extract$datasetname - ERDDAP dataset name
- extract$tlong - the longitudes on some scale as request
- extract$tlatitude - the latitudes always going south to north
- extract$ttime - the times of the extracts

**Details**

rxtractogon extracts the data from the smallest bounding box that contains the polygon, and then uses the function "point.in.polygon" from the "sp" package to mask out the areas outside of the polygon. rxtractogon only works with datasets defined on a latitude and longitude grid.

**Examples**

```r
# toy example to show use
# and keep execution time low

dataInfo <- rerddap::info('erdHadISST')

parameter <- 'sst'
tcoord <- c("2016-06-15")
xcoord <- mbnms$Longitude[1:3]
ycoord <- mbnms$Latitude[1:3]
sanctSST <- rxtractogon (dataInfo, parameter=parameter, xcoord = xcoord, ycoord = ycoord, tcoord= tcoord)

xcoord <- mbnms$Longitude
ycoord <- mbnms$Latitude
dataInfo <- rerddap::info('etopo180')
parameter = 'altitude'
xName <- 'longitude'
yName <- 'latitude'
bathy <- rxtractogon (dataInfo, parameter = parameter, xcoord = xcoord, ycoord = ycoord)
```

**Description**

*rxtracto_3D* uses the R program 'rerddap' to extract environmental data from an 'ERDDAP' server in an (x,y,z, time) bounding box. The same call could be made directly in rerddap, but function is maintained as it is used in the polygon routine.
Usage

`rxtracto_3D(
  dataInfo,
  parameter = NULL,
  xcoord = NULL,
  ycoord = NULL,
  zcoord = NULL,
  tcoord = NULL,
  xName = "longitude",
  yName = "latitude",
  zName = "altitude",
  tName = "time",
  verbose = FALSE,
  cache_remove = TRUE
)
``

Arguments

dataInfo - the return from an 'rerddap:info' call to an 'ERDDAP' server
parameter - character string containing the name of the parameter to extract
xcoord - a real array with the x-coordinates of the trajectory (if longitude in #' decimal degrees East, either 0-360 or -180 to 180)
ycoord - a real array with the y-coordinate of the trajectory (if latitude in decimal degrees N; -90 to 90)
zcoord - a real array with the z-coordinate (usually altitude or depth)
tcoord - a character array with the times of the trajectory in "YYYY-MM-DD" - for now restricted to be time.
xName - character string with name of the xcoord in the 'ERDDAP' dataset (default "longitude")
yName - character string with name of the ycoord in the 'ERDDAP' dataset (default "latitude")
zName - character string with name of the zcoord in the 'ERDDAP' dataset (default "altitude")
tName - character string with name of the tcoord in the 'ERDDAP' dataset (default "time")
verbose - logical variable (default FALSE) if the the URL request should be verbose
cache_remove - logical variable (default TRUE) whether to delete 'rerddap' cache

Value

structure with data and dimensions:

- extract$data - the data array dimensioned (lon,lat,time)
- extract$varname - the name of the parameter extracted
- extract$datasetname - ERDDAP dataset name
• extract$longitude - the longitudes on some scale as request
• extract$latitude - the latitudes always going south to north
• extract$time - the times of the extracts

Examples

# toy example to show use
# and keep execution time low

dataInfo <- rerddap::info('erdHadISST')

parameter <- 'sst'
xcoord <- c(-130.5, -130.5)
ycoord <- c(40.5, 40.5)
tcoord <- c('2006-01-16', '2006-01-16')
exttract <- rxtracto_3D(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord,
tcoord = tcoord)

# 2-D example getting bathymetry
dataInfo <- rerddap::info('etopo360')
parameter <- 'altitude'
extrack <- rxtracto_3D(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord)

swchl

swchl Data

Description

pre-Download of Pacific West Coast SST fro use in ‘plotTrack’ example can run within CRAN

Usage

swchl

Format

An object of class list (inherits from rxtractoTrack) of length 13.

Details

obtained using the ‘rxtracto’ command
tagData <- Marlintag38606 xpos <- tagData$lon[1:20]
ypos <- tagData$lat[1:20] tpos <- tagData$date[1:20] tpos <- tagData$date[1:20] zpos <- rep(0., length(xpos)) swchlInfo <- rerddap::info('erdSWchla8day') swchl <- rxtracto(swchlInfo, parameter = 'chlorophyll', xcoord = xpos, ycoord = ypos, tcoord = tpos, zcoord = zpos, xlen = .2, ylen = .2)
Index

* datasets
  dataInfo, 2
  Marlintag38606, 2
  mbnms, 3
  MBsst, 4
  swchl, 13

dataInfo, 2
Marlintag38606, 2
mbnms, 3
MBsst, 4

plotBBox, 4
plotTrack, 5

rerddapXtracto, 7
rxtracto, 7
rxtracto_3D, 11
rxtractogon, 10

swchl, 13