

# Package ‘rerddapXtracto’

June 13, 2020

**Type** Package

**Title** Extracts Environmental Data from 'ERDDAP' Web Services

**Version** 0.4.8

**Date** 2020-06-11

**Description** Contains three functions that access environmental data from any 'ERDDAP' data web service. The `rextracto()` function extracts data along a trajectory for a given "radius" around the point. The `rextracto_3D()` function extracts data in a box. The `rextractogon()` 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(>= 3.6.0)

**License** CC0

**Imports** abind, dplyr, ggplot2, httr, maps, methods, ncd4, parsedate, plotdap (>= 0.0.5), readr, rerddap (>= 0.6.0), sf, sp, stats, utils

**Suggests** cmocean, gganimate, knitr, mapdata, maptools, rgdal, rmarkdown

**RoxygenNote** 7.1.0

**Encoding** UTF-8

**LazyData** TRUE

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Roy Mendelsohn [aut, cre]

**Maintainer** Roy Mendelsohn <roy.mendelsohn@noaa.gov>

**Repository** CRAN

**Date/Publication** 2020-06-12 22:10:02 UTC

**R topics documented:**

dataInfo . . . . .	2
Marlintag38606 . . . . .	2
mbnms . . . . .	3
MBSst . . . . .	4
plotBBox . . . . .	4
plotTrack . . . . .	5
rerddapXtracto . . . . .	7
rextracto . . . . .	7
rextractogon . . . . .	9
rextracto_3D . . . . .	11
swchl . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

dataInfo	<i>dataInfo Data</i>
----------	----------------------

---

**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	<i>Marlin Tag Data</i>
----------------	------------------------

---

**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

**highLon** high error on longitude

**lowLat** low error on latitude

**highLat** 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**

[http://sanctuaries.noaa.gov/library/imast\\_gis.html](http://sanctuaries.noaa.gov/library/imast_gis.html)

---

 MBSst

*MBSst Data*


---

### Description

pre-Download of Pacific West Coast SST fro 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 `dataInfo <- rerddap::info('erdMBSstd1day')` 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

plotBox is a function to plot the results from 'rxtracto()' and 'xtracto()'

### Usage

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

**Arguments**

resp	data frame returned from 'rextracto()'
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**

```
dataInfo <- rerddap::info('erdMBsstd1day')
parameter <- 'sst'
xcoord <- c(230, 232)
ycoord <- c(33, 35)
tcoord <- c('2006-01-15', '2006-01-15')
zcoord <- c(0., 0.)
MBsst <- rextracto_3D(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord,
                     tcoord = tcoord, zcoord = zcoord)

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

---

plotTrack

*plot result of 'rextracto'*

---

**Description**

plotTrack is a function to plot the results from 'rextracto()'

**Usage**

```
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

resp	data frame returned from 'rxtracto()'
xcoord	passed to 'rxtracto()'
ycoord	passed to 'rxtracto()'
tcoord	passed to 'rxtracto()'
plotColor	the color to use in plot from 'cmocean'
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
shape	shape to use to mark track
size	size of shape to use to mark track

### Value

a 'plotdap' plot

### Examples

```

tagData <- Marlintag38606
xpos <- tagData$lon[1:20]
ypos <- tagData$lat[1:20]
tpos <- tagData$date[1:20]
zpos <- tagData$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)

p <- plotTrack(swchl, xpos, ypos, tpos, plotColor = 'algae')

```

---

rerddapXtracto	<i>rerddapXtracto: Routines to simplify data extraction using ERD's ERDDAP web service.</i>
----------------	---

---

## 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. # @section Details: 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	<i>Extract environmental data along a trajectory from an 'ERDDAP' server using 'rerddap'.</i>
----------	---

---

## Description

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

## Usage

```
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",
    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 degrees 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")
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 <- rxttracto(dataInfo, parameter = parameter, xcoord = xcoord,
                    ycoord = ycoord, tcoord= tcoord,
                    xlen = xlen, ylen = ylen)

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

---

rxttractogon

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

---

### Description

rxttractogon uses the R program 'rerddap' to extract environmental data from an 'ERDDAP' server in a polygon through time.

**Usage**

```

rextractogon(
  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 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 masked 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

## 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

```
# 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)
```

---

rxtracto\_3D

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

---

## 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')
extract <- rxtracto_3D(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord,
                       tcoord = tcoord)

# 2-D example getting bathymetry
dataInfo <- rerddap::info('etopo360')
parameter <- 'altitude'
extract <- rxtracto_3D(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord)
# Dataset that has depth also
# 3 months of subsurface temperature at 70m depth from SODA 2.2.4
dataInfo <- rerddap::info('erdSoda331oceanmday')
parameter = 'temp'
xName <- 'longitude'
yName <- 'latitude'
zName <- 'depth'
xcoord <- c(230.25, 250.25)
ycoord <- c(30.25, 43.25)
zcoord <- c(5.03355, 15.10065)
tcoord <- c('2010-10-15', '2010-12-15')
extract <- rxtracto_3D(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord,
                       zcoord = zcoord, tcoord = tcoord, xName = xName,
                       yName = yName, zName = zName)

```

---

swchl

*swchl Data*


---

**Description**

pre-Download of Pacific West Coast SST fro use in 'plotTrack()' example can run within CRAN Time limits

**Usage**

```
swchl
```

**Format**

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

**Details**

obtained using the 'rextracto()' 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 <- rextracto(swchlInfo, parameter`  
`= 'chlorophyll', xcoord = xpos, ycoord = ypos, tcoord = tpos, zcoord = zpos, xlen = .2, ylen = .2)`

# Index

## \*Topic **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](#)

rextracto, [7](#)

rextracto\_3D, [11](#)

rextractogon, [9](#)

swchl, [13](#)