

# Package ‘rerddapXtracto’

February 15, 2026

**Type** Package

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

**Version** 1.2.5

**Date** 2026-02-14

**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(>= 4.0.0)

**License** CC0

**Imports** abind, dplyr, ggplot2, httr, maps, methods, ncd4, parsedate, plotdap (>= 1.1.1), readr, rerddap (>= 1.2.2), sf, sp, stats, tibble, utils

**Suggests** cmocean, gganimate, knitr, lubridate, mapdata, rmarkdown

**RoxygenNote** 7.3.3

**Encoding** UTF-8

**LazyData** TRUE

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Roy Mendelsohn [aut, cre],  
Marie Auger-Methe [ctb],  
Sunny Bak-Hospital [ctb]

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

**Repository** CRAN

**Date/Publication** 2026-02-15 18:30:09 UTC

## Contents

dataInfo . . . . .	2
Marlintag38606 . . . . .	3
mbnms . . . . .	3
MBSst . . . . .	4
PB_Argos . . . . .	4
plotBBox . . . . .	5
plotTrack . . . . .	6
rxtracto . . . . .	7
rxtractogon . . . . .	10
rxtracto_3D . . . . .	11
safe_info . . . . .	13
swchl . . . . .	14
tidy_grid . . . . .	14
<b>Index</b>	<b>16</b>

---

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

**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**

[https://sanctuaries.noaa.gov/library/imast\\_gis.html](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 `rextracto3D`) of length 6.

**Details**

obtained using the `rextracto_3D()` command `dataInfo <- rerddap::info('erdMBsst1day')` `parameter <- 'sst'` `xcoord <- c(230, 230.1)` `ycoord <- c(33, 33.1)` `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)`

---

PB\_Argos

*Polar Bear Track Data*

---

**Description**

Telemetry data of a tagged polar bear

**Usage**

PB\_Argos

**Format**

A data frame with 1919 obs. of 4 variables:

**DateTime** time of observation

**QualClass** Quality Flag

**Lat** latitude of observation

**Lon** longitude of observation

**Source**

**dryad dataset** <https://datadryad.org/stash/dataset/doi:10.5061/dryad.4qrfj6q96>

**Citation** Auger-Méthé, Marie; Derocher, Andrew E. (2021). Argos and GPS data for a polar bear track [Dataset]. Dryad. <https://doi.org/10.5061/dryad.4qrfj6q96>

**Associated Paper** Auger-Méthé, M., Newman, K., Cole, D., Empacher, F., Gryba, R., King, A. A., Leos-Barajas, V., Mills Flemming, J., Nielsen, A., Petris, G., and Thomas, L.. 2021. A guide to state–space modeling of ecological time series. *Ecological Monographs* 91(4):e01470. [10.1002/ecm.1470](https://doi.org/10.1002/ecm.1470)

---

plotBBox

*plot result of 'rtracto\_3D'*

---

**Description**

plotBBox is a function to plot the results from 'rtracto\_3D()' and 'rtractogon()'

**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 'rtracto_3D()' or 'rtractogon()'
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**

```
## example code to download data for plotBBox
## dataInfo <- rerddap::info('erdMBSstd1day')
## parameter <- 'sst'
## xcoord <- c(230, 230.1)
## ycoord <- c(33, 33.1)
## 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, maxpixels = 50))
```

---

plotTrack

*plot result of 'rxtracto'*

---

**Description**

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

**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 'rextracto()'
tcoord	passed to 'rextracto()'
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

```
## example data download for plotTrack
## tagData <- Marlintag38606
## xpos <- tagData$lon[1:20]
## ypos <- tagData$lat[1:20]
## tpos <- tagData$date[1:20]
## zpos <- rep(0., length(xpos))

## example data download for plotTrack
## swchlInfo <- rerddap::info('erdSWchla8day')
##scwchl <- rextracto(swchlInfo, parameter = 'chlorophyll', xcoord = xpos,
##                    ycoord = ypos, tcoord = tpos, zcoord = zpos, xlen = .2, ylen = .2)
##
## suppressWarnings(p <- plotTrack(swchl, xpos, ypos, tpos, plotColor = 'algae'))
```

---

rextracto	<i>Extract environmental data along a trajectory from an 'ERDDAP™' server using 'rerddap'.</i>
-----------	--

---

### Description

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

**Usage**

```

rextracto(
  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 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")

interp - array (size 2) of character strings - c(interpolation type, neighborhood) Uses the new ERDDAP™ interpolation option to get values See Vignette for details Default 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

If success 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

else an error string

### 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')
# extract <- rextracto(dataInfo, parameter = parameter, xcoord = xcoord,
#                       ycoord = ycoord, tcoord= tcoord
#                       )
##
## bathymetry example
## 2-D example getting bathymetry
## Wrap rerddap::info('etopo360') call in function that insures proper finish if it fails
dataInfo <- safe_info('etopo360')
parameter <- 'altitude'
# extract <- rextracto(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord)

```

---

rextractogon	<i>Extract environmental data in a polygon using 'ERDDAP™' and 'rerddap'.</i>
--------------	---

---

### Description

rextractogon 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**

If successful a 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

else an error string

**Details**

`rextractogon` 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. `rextractogon` 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 <- rextractogon (dataInfo, parameter=parameter, xcoord = xcoord,
#                           ycoord = ycoord, tcoord= tcoord)
#
## MBMS bathymetry example
xcoord <- mbnms$Longitude
ycoord <- mbnms$Latitude
## Wrap rerddap::info('etopo180') call in function that insures proper finish if it fails
dataInfo <- safe_info('etopo180')
parameter = 'altitude'
xName <- 'longitude'
yName <- 'latitude'
# bathy <- rextractogon (dataInfo, parameter = parameter, xcoord = xcoord, ycoord = ycoord)
```

---

`rextracto_3D`

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

---

**Description**

`rextracto_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**

```

rextracto_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**

If successful a 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

else an error string

## 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)

## bathymetry example
## 2-D example getting bathymetry
## Wrap rerddap::info('etopo360') call in function that insures proper finish if it fails
dataInfo <- safe_info('etopo360')
parameter <- 'altitude'
# extract <- rxtracto_3D(dataInfo, parameter, xcoord = xcoord, ycoord = ycoord)
```

---

safe_info	<i>Wrap rerddap::info() so that it will fail gracefully safe_info uses tryCatch() to deal with unavailable response</i>
-----------	---

---

## Description

Wrap rerddap::info() so that it will fail gracefully safe\_info uses tryCatch() to deal with unavailable response

## Usage

```
safe_info(dataset_id, url = "https://upwell.pfeg.noaa.gov/erddap/")
```

## Arguments

dataset_id	ERDDAP dataset_id
url	base ERDDAP URL

## Value

If success, dataset information, otherwise message and end

**Examples**

```
dataInfo <- safe_info('etopo360')
```

---

swchl	<i>swchl Data</i>
-------	-------------------

---

**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 'rxtracto()' command 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)

---

tidy_grid	<i>convert result of 'rxtracto_3D' or 'rxtractogon' to tidy long-format</i>
-----------	---

---

**Description**

tidy\_grid is a function to convert result of 'rxtracto\_3D' or 'rxtractogon' to "tidy" long-format

**Usage**

```
tidy_grid(response, as_tibble = TRUE)
```

**Arguments**

response	data frame returned from 'rxtracto_3D()' or 'rxtractogon()'
as_tibble	whether to return as tibble or dataframe

**Value**

a dataframe in long-format

**Examples**

```
MBsst_tidy <- tidy_grid(MBsst)
```

# Index

## \* datasets

- dataInfo, [2](#)
- Marlintag38606, [3](#)
- mbnms, [3](#)
- MBsst, [4](#)
- PB\_Argos, [4](#)
- swchl, [14](#)

dataInfo, [2](#)

Marlintag38606, [3](#)

mbnms, [3](#)

MBsst, [4](#)

PB\_Argos, [4](#)

plotBBox, [5](#)

plotTrack, [6](#)

rxtracto, [7](#)

rxtracto\_3D, [11](#)

rxtractogon, [10](#)

safe\_info, [13](#)

swchl, [14](#)

tidy\_grid, [14](#)