Package 'satin'

October 14, 2022

Type Package

Title Visualisation and Analysis of Ocean Data Derived from Satellites

Version 1.1.0

Date 2022-09-22

Encoding UTF-8

Author Héctor Villalobos and Eduardo González-Rodríguez <egonzale@cicese.mx>

Maintainer Héctor Villalobos <hvillalo@ipn.mx>

Description

With 'satin' functions, visualisation, data extraction and further analysis like producing climatologies from several images, and anomalies of satellite derived ocean data can be easily done. Reading functions can import a user defined geographical extent of data stored in netCDF files. Currently supported ocean data sources include NASA's Ocean-color web page <https://oceancolor.gsfc.nasa.gov/>, sensors VIIRS-SNPP; MODIS-Terra; MODIS-Aqua; and SeaWiFS. Available variables from this source includes chlorophyll concentration, sea surface temperature (SST), and several others. Data sources specific for SST that can be imported too includes Pathfinder AVHRR <https://www.ncei.noaa.gov/products/avhrr-pathfinder-sst> and GHRSST <https://www.ghrsst.org/>. In addition, ocean productivity data produced by Oregon State University <http://sites.science.oregonstate.edu/ocean.productivity/> can also be handled previous conversion from HDF4 to HDF5 format. Many other ocean variables can be processed by importing netCDF data files from two European Union's Copernicus Marine Service databases <https://marine.copernicus.eu/>, namely Global Ocean Physical Reanalysis and Global Ocean Biogeochemistry Hindcast.

Depends R (>= 3.5.0)

License GPL-3

URL https://github.com/hvillalo/satin

BugReports https://github.com/hvillalo/satin/issues

Collate satin-class.R read.cmems.R read.ghrsst.R read.nasaoc.R read.osunpp.R fmainPos.R imageScale.R plot.satin.R satinPalette.R quiver.R verticalProfiles.R satinDataframe.R pixelate.R climatology.R crop.R extractPts.R isolines.R satinMean.R anomaly.R velocity.R

Imports ncdf4, maps, PBSmapping, grDevices, splancs, sp, geosphere, graphics, methods

NeedsCompilation no

Repository CRAN

Date/Publication 2022-09-22 21:00:02 UTC

R topics documented:

`anomaly` *Anomalies of a satin object*

Description

This function returns ocean data anomalies of a satin object. Anomalies are calculated subtracting a conforming average satin object.

Usage

```
anomaly(X, Y)
```

Arguments

X	a satin object obtained with satinMean for which anomalies are to be calculated. It must have identifying labels in the slot attribs.
Y	also a satin object returned by satinMean with the means to be subtracted from X. It must have the same labels as X.

Details

The anomalies of a satin object represent, for every pixel, the values above and below a reference average. A typical example can be monthly sea surface temperature anomalies for various years. These are calculated subtracting the corresponding average month for the whole period (see example below).

In principle, anomalies can be calculated for other ocean variables and time periods, but is up to the user to determine if that makes sense. For example, let say that Z is a satin object with daily chlorophyll-a concentration data for several years. In order to calculate quarterly anomalies, we will need first to obtain X <- satinMean(Z, "%Y-%qtr") and Y <- satinMean(Z, "%qtr").

Value

An object of class "satin" (see satin-class for details) where the third dimension in the data array accomodates the calculated anomalies. An extra element (labels) is included in the slot attribs to identify the time period of the anomalies.

Author(s)

Héctor Villalobos and Eduardo González-Rodríguez

See Also

satinMean for calculating ocean data averages needed, and climatology for climatologies.

Examples

```
if(interactive()){
# Calculate monthly sea surface temperature from weekly data.
# sst contains 240 weekly images for five years.

# This will produce the 12 average months for the five years
sst.m <- satinMean(sst, by="%m")

# and here we have 60 monthly periods, 12 for each one of the five years
sst.ym <- satinMean(sst, by="%Y-%m")

# monthly sea surface temperature anomalies
anom <- anomaly(X = sst.ym, Y = sst.m)

# the dimensions of the data slots in sst.ym and anom are the same.
dim(sst.ym@data); dim(anom@data)

# plots of the first 24 months
```

```
lab <- paste(month.name, rep(2014:2018, each=12))
for (m in 1:24){
  plot(anom, period = m, zlim=c(-10, 10), col.sep = 0.2, main = lab[m],
       scheme = c("blue", "cyan", "white", "yellow", "red"))
}
}
```

`climatology` *Climatology of a satin object*

Description

This function calculates the percent coverage, mean, standard deviation, minimum and maximum for every pixel in a set of images stored in a single satin data object.

Usage

```
climatology(X, depth = NULL)
```

Arguments

`X`	a satin object as returned by `read.nasaoc`, `read.ghrsst`, `read.osunpp`, or `read.cmems`.
`depth`	an integer referring to a given depth level. Only meaningful for Copernicus data that includes variables at different depths.

Details

For Copernicus data files containing more than one variable, the `read.cmems` function returns a named list, with elements corresponding to each variable. In order to use these objects with `climatology` the appropriate indexing should be applied (see examples below).

Value

An object of class `"satin"` (see `satin-class` for details) where the third dimension in the data array accomodates coverage, mean, standard deviation, minimum and maximum. Also, an extra element (`label`) is included in the slot `attribs` to summarize the start and end times used for the climatology.

Author(s)

Héctor Villalobos and Eduardo González-Rodríguez

Examples

```
if(interactive()){
csst <- climatology(sst) # sst is a satin object with 12 monthly images
                         # of sea surface temperature

# -- plots --

# coverage
plot(csst, period = 1)

# mean
plot(csst, period = 2)

# standard deviation
plot(csst, period = 3)

# minimum
plot(csst, period = 4)

# maximum
plot(csst, period = 5)
}
# For Copernicus data

data(dcmems) # load sample data
names(dcmems) # available variables

# mean potential temperature (thetao) at 0.49 m
# using labels stored in slot period
cthetao1 <- climatology(dcmems$thetao, depth = 1)
plot(cthetao1, period = 2, main = cthetao1@period$label[2])

# standard deviation of thetao at 318 m
cthetao2 <- climatology(dcmems$thetao, depth = 5)
plot(cthetao2, period = 3, main = cthetao1@period$label[3])
```

---

crop *Cut a satin object*

Description

Select and cut a polygon shaped area of interest from a satin object.

Usage

```
crop(X, polygon = NULL, return.poly = FALSE)
```

Arguments

X	a satin object as returned by read.nasaoc, read.ghrsst, read.osunpp, or read.cmems.
polygon	a polygon defining the area of interest (aoi) to be cutted.
return.poly	if TRUE, crop returns a list with the cutted satin object and the polygon drawn (a data frame).

Details

If provided, the polygon defining the aoi must have two columns with x and y coordinates. The function verifies that the last point is exactly the same as the first one. When not specified, the user can draw a polygon by clicking on a previously displayed image. Note that for drawing the polygon in this manner, the plot of the satin object has to be created using colbar = FALSE. The result is a satin object where the pixels outside the aoi are set to NA. In order to cut Copernicus data returned from read.cmems, the appropriate indexing should be previously done.

Value

A cutted satin object.

When return.poly = TRUE, a list with two components:

aoi	cutted satin object.
polygon	polygon coordinates defining the aoi.

Author(s)

Héctor Villalobos

See Also

getpoly

Examples

```
# load and display sample data
data(dsst)
plot(dsst)

# define polygon, cut the image and display the aoi
polyg <- data.frame(x = c(-110, -105, -113, -116, -110), y = c(23, 23, 32, 32, 23))
dsst.clip <- crop(dsst, polygon = polyg)
plot(dsst.clip)

if(interactive()){
# the same, but polygon is defined by clicking on the image
plot(dsst, colbar = FALSE)
dsst.clip2 <- crop(dsst, return.poly = TRUE) # *** draw the aoi polygon ***
dsst.clip2$polygon # show polygon coordinates
plot(dsst.clip2$aoi) # plot the selected aoi
}
```

dchl *Chlorophyll concentration sample data - Aqua Modis sensor*

Description

Chlorophyll-a concentration (8-day average; 4 km resolution) from March 30 to April 7 2013 off northwest Mexico.

Usage

```
data("dchl")
```

Format

Formal class 'satin' [package "satin"] with 6 slots

```
..@ lon                 : num [1:360] -119 -119 -119 -119 -119 ...
..@ lat                 : num [1:360] 20 20.1 20.1 20.1 20.2 ...
..@ data                : num [1:360, 1:360, 1] 0.0631 0.0696 0.0724 0.0723 0.0705 ...
..@ attribs: List of 6
.. ..$ title            : chr "HMODISA Level-3 Standard Mapped Image"
.. ..$ longname         : chr "Chlorophyll Concentration, OCI Algorithm"
.. ..$ name             : chr "chlor_a"
.. ..$ units            : chr "mg m^-3"
.. ..$ temporal_range   : chr "8-day"
.. ..$ spatial_resolution : chr "4.64 km"
..@ period: List of 2
.. ..$ tmStart          : POSIXct[1:1], format: "2013-03-30 00:25:01"
.. ..$ tmEnd            : POSIXct[1:1], format: "2013-04-07 02:59:59"
..@ depth               : num(0)
```

Details

Source data file (A20130892013096.L3m_8D_CHL_chlor_a_4km.nc) was downloaded from the link below, then imported with the `read.nasaoc` function. The selected area covers from 20 to 35 degrees of latitude North and from 119 to 104 degrees of longitude West.

Source

https://oceancolor.gsfc.nasa.gov/cgi/l3

References

https://oceancolor.gsfc.nasa.gov/docs/technical/

Examples

```
data(dchl)
```

```
dchl
str(dchl)

plot(dchl)

# plot with logarithmic scale
plot(dchl, log = TRUE)
```

---

dcmems *Copernicus Marine Service sample data*

Description

This sample data contains monthly values (November and December 2010) of sea water potential temperature (thetao, degrees C) and sea surface height (zos, m) off northwest Mexico, 1/12 degree horizontal resolution (aprox. 9.26 km).

Usage

```
data("dcmems")
```

Format

List of 2
$ thetao:Formal class 'satin' [package "satin"] with 6 slots

```
.. ..@ lon                : num [1:181] -119 -119 -119 -119 -119 ...
.. ..@ lat                : num [1:169] 20 20.1 20.2 20.2 20.3 ...
.. ..@ data               : num [1:169, 1:181, 1:2, 1:5] 23.5 23.5 23.5 23.4 23.4 ...
.. ..@ attribs:List of 6
.. .. ..$ title           : chr "thetao"
.. .. ..$ longname        : chr "Temperature"
.. .. ..$ name            : chr "thetao"
.. .. ..$ units           : chr "degrees_C"
.. .. ..$ temporal_range  : chr "monthly"
.. .. ..$ spatial_resolution : chr "9.2 km"
.. ..@ period :List of 2
.. .. ..$ tmStart         : POSIXct[1:2], format: "2010-11-16 00:00:00" ...
.. .. ..$ tmEnd           : POSIXct[1:2], format: "2010-11-16 00:00:00" ...
.. ..@ depth              : num [1:5] 0.494 9.573 29.445 92.326 318.127
```

$ zos :Formal class 'satin' [package "satin"] with 6 slots

```
.. ..@ lon                : num [1:181] -119 -119 -119 -119 -119 ...
.. ..@ lat                : num [1:169] 20 20.1 20.2 20.2 20.3 ...
.. ..@ data               : num [1:169, 1:181, 1:2, 1] 0.336 0.34 0.343 0.345 0.347 ...
.. ..@ attribs:List of 6
.. .. ..$ title           : chr "zos"
```

```
.. .. ..$ longname            : chr "Sea surface height"
.. .. ..$ name                : chr "zos"
.. .. ..$ units               : chr "m"
.. .. ..$ temporal_range      : chr "monthly"
.. .. ..$ spatial_resolution  : chr "9.2 km"
.. ..@ period :List of 2
.. .. ..$ tmStart             : POSIXct[1:2], format: "2010-11-16 00:00:00" ...
.. .. ..$ tmEnd               : POSIXct[1:2], format: "2010-11-16 00:00:00" ...
.. ..@ depth                  : num 0.494
```

Details

Source data corresponds to Global Ocean Physical Reanalysis database (GLOBAL_REANALYSIS_PHY_001_030), and was downloaded from the link below, then imported with the `read.cmems` function. The selected area covers from 20 to 35 degrees of latitude North and from 119 to 104 degrees of longitude West. Potential temperature includes five depth levels (0.494, 9.573, 29.445, 92.326, 318.127).

Source

https://marine.copernicus.eu/

References

Fernandez, E. and Lellouche, J. M. 2018 *Product Manual for the Global Ocean Physical Reanalysis product, issue 1.1.* EU Copernicus Marine Service.

Examples

```
data(dcmems)
dcmems

plot(dcmems$thetao)

# potential temperature at 318 m
plot(dcmems$thetao, depth = 5)

# sea surface height
plot(dcmems$zos)
```

---

dmap *Map of northwest Mexico*

Description

Map of northwest Mexico obtained from GSHHG database (v2.3.7).

Usage

```
data("dmap")
```

Details

This intermediate resolution map was imported from the Global Self-consistent, Hierarchical, High-resolution Geography Database by means of `Rgshhs` function from **maptools** package.

Source

https://www.soest.hawaii.edu/pwessel/gshhg/gshhg-bin-2.3.7.zip

References

Wessel, P. and Smith, W. H. F. 1996 A Global Self-consistent, Hierarchical, High-resolution Shoreline Database, *Journal of Geophysical Research*, **101**, 8741–8743.

See Also

`Rgshhs`

Examples

```
library(sp)
data(dmap)
plot(dmap, xlim=c(-120, -105) , ylim=c(20, 35), xaxs="i", yaxs="i",
     axes=TRUE, col="beige", lty=1, border="grey70"); box()
```

---

dnpp *Ocean Net Primary Production sample data*

Description

Ocean Net Primary Production (8-day average; 9.26 km resolution) from March 30 to April 6 2013 off northwest Mexico.

Usage

```
data("dnpp")
```

Format

Formal class 'satin' [package "satin"] with 6 slots

```
..@ lon                 : num [1:180] -119 -119 -119 -119 -119 ...
..@ lat                 : num [1:180] 20.1 20.1 20.2 20.3 20.4 ...
..@ data                : num [1:180, 1:180, 1] 285 292 270 279 262 ...
..@ attribs:List of 6
.. ..$ title            : chr "Ocean Productivity"
.. ..$ longname         : chr "Ocean Net Primary Production"
.. ..$ name             : chr "npp"
.. ..$ units            : chr "mgC m-2 day-1"
```

```
.. ..$ temporal_range      : chr "8 day"
.. ..$ spatial_resolution  : chr "9.26 km"
..@ period :List of 2
.. ..$ tmStart             : POSIXct[1:1], format: "2013-03-30"
.. ..$ tmEnd               : POSIXct[1:1], format: "2013-04-06 23:59:59"
..@ depth                  : num(0)
```

Details

Source data file (vgpm.2013089.hdf.gz) was downloaded from the link below, decompressed with `gunzip` function from **R.utils** package, then converted to hdf v5 with h4toh5 Conversion Software (`https://support.hdfgroup.org/products/hdf5_tools/h4toh5/download.html`) before being imported with the `read.osunpp` function. The selected area covers from 20 to 35 degrees of latitude North and from 119 to 104 degrees of longitude West.

Source

`http://sites.science.oregonstate.edu/ocean.productivity/standard.product.php`

References

Behrenfeld, M. J. and Falkowski, P. G. 1997 Photosynthetic rates derived from satellite-based chlorophyll concentration, *Limnology and Oceanography*, **42**, 1–20.

Examples

```
data(dnpp)
dnpp
str(dnpp)
plot(dnpp, col.sep = 100)
```

`dsst` *SST sample data - Aqua Modis sensor*

Description

Sea Surface Temperature (4 um nighttime; 8-day average; 4 km resolution) from March 29 to April 6 2013 off northwest Mexico.

Usage

```
data("dsst")
```

Format

Formal class 'satin' [package "satin"] with 6 slots

```
..@ lon                 : num [1:360] -119 -119 -119 -119 -119 ...
..@ lat                 : num [1:360] 20 20.1 20.1 20.1 20.2 ...
..@ data                : num [1:360, 1:360, 1] NA NA NA NA 18.2 ...
..@ attribs :List of 6
.. ..$ title            : chr "HMODISA Level-3 Standard Mapped Image"
.. ..$ longname         : chr "4um Sea Surface Temperature"
.. ..$ name             : chr "sst4"
.. ..$ units            : chr "degree_C"
.. ..$ temporal_range   : chr "8-day"
.. ..$ spatial_resolution : chr "4.64 km"
..@ period :List of 2
.. ..$ tmStart          : POSIXct[1:1], format: "2013-03-29 12:05:08"
.. ..$ tmEnd            : POSIXct[1:1], format: "2013-04-06 14:50:08"
..@ depth               : num(0)
```

Details

Source data file (A20130892013096.L3m_8D_SST4_sst4_4km.nc) was downloaded from the link below, then imported with the `read.nasaoc` function. The selected area covers from 20 to 35 degrees of latitude North and from 119 to 104 degrees of longitude West.

Source

https://oceancolor.gsfc.nasa.gov/cgi/l3

References

https://oceancolor.gsfc.nasa.gov/docs/technical/

Examples

```
data(dsst)
dsst
str(dsst)
plot(dsst)
```

`extractPts` *Extract ocean data values from a satin object*

Description

Extract data values for pixels selected from an image of a satin object.

Usage

```
extractPts(X, points = NULL)
```

Arguments

X	a satin object as returned by read.nasaoc, read.ghrsst, read.osunpp, or read.cmems.
points	a data frame with point(s) longitude and latitude coordinates where to extract data values.

Details

If provided, points must have two columns with x and y coordinates. When not specified, points can be selected by clicking on the image. Note that for selecting the points in this manner, the plot of the satin object has to be created using colbar = FALSE.

Value

A data frame with at least 7 columns: For each point, a sequential id number; (x, y) coordinates, either specified via the points argument or selected by clicking on the image; (lon and lat) coordinates of pixels closer to the desired points; and for quality control purposes, the distance between them (d, km). The remaining column(s) are the parameter values for every image in the satin object.

Author(s)

Héctor Villalobos

Examples

```
# load data, define coordinates and extract values
data(dsst)

plot(dsst)
coord <- data.frame(x = seq(-130, -114, 2), y = seq(20, 36, 2))
values <- extractPts(dsst, points = coord)
values

# if no points are given the user can select them by clicking on the image
if (interactive()){
plot(dsst, colbar = FALSE)
values2 <- extractPts(dsst) # *** click on the image ***
values2
}
```

fmainPos *Calculate coordinates for title in plots*

Description

Internal function called by `plot.satin` and `quiver`.

Usage

```
fmainPos(pu, main.pos)
```

Arguments

`pu`	Obtained by par("usr").
`main.pos`	Desired position, one among "topright", "topleft", "bottomright", "bottomleft").

Author(s)

Héctor Villalobos

imageScale *Make a color scale to accompany an image or other plot*

Description

The `imageScale` function is wrapper for `imageScale` and accepts the same arguments. It converts a vector of values (`z`) to a vector of color levels. One must define the number of colors. The limits of the color scale ("zlim") or the break points for the color changes("breaks") can also be defined. When breaks and zlim are defined, breaks overrides zlim. All arguments are similar to those in the `image` function. Appearance is best when incorporated with `layout`.

Usage

```
imageScale(z, zlim, col = heat.colors(12), breaks, axis.pos = 1,
  add.axis = TRUE, xlim = NULL, ylim = NULL, log, ...)
```

Arguments

`z`	A vector or matrix of values.
`zlim`	Limits of the color scale values.
`col`	Vector of color values (default is 12 colors from the `heat.colors` palette).
`breaks`	Break points for color changes. If breaks is specified then `zlim` is unused and the algorithm used follows `cut`, so intervals are closed on the right and open on the left except for the lowest interval which is closed at both ends.

axis.pos	Position of the axis (1=bottom, 2=left, 3=top, 4=right) (default = 1).
add.axis	Logical (TRUE/FALSE). Defines whether the axis is added (default: TRUE).
xlim	Limits for the x-axis.
ylim	Limits for the y-axis.
log	logical. If TRUE, z is log transformed (base 10).
...	Additional graphical parameters to pass to the image function.

Author(s)

Marc Taylor, modified for log scale by Héctor Villalobos

Examples

```
# Make color palettes
pal.1=colorRampPalette(c("green4", "orange", "red", "white"), space="rgb", bias=0.5)
pal.2=colorRampPalette(c("blue", "cyan", "yellow", "red", "pink"), space="rgb")

# Make images with corresponding scales
op <- par(no.readonly = TRUE)
layout(matrix(c(1,2,3,0,4,0), nrow=2, ncol=3), widths=c(4,4,1), heights=c(4,1))
#layout.show(4)
#1st image
breaks <- seq(min(volcano), max(volcano),length.out=100)
par(mar=c(1,1,1,1))
image(seq(dim(volcano)[1]), seq(dim(volcano)[2]), volcano,
col=pal.1(length(breaks)-1), breaks=breaks-1e-8, xaxt="n", yaxt="n", ylab="", xlab="")
#Add additional graphics
levs <- pretty(range(volcano), 5)
contour(seq(dim(volcano)[1]), seq(dim(volcano)[2]), volcano, levels=levs, add=TRUE)
#Add scale
par(mar=c(3,1,0,1))
imageScale(volcano, col=pal.1(length(breaks)-1), breaks=breaks-1e-8,axis.pos=1)
abline(v=levs)
box()

#2nd image
breaks <- c(0,100, 150, 170, 190, 200)
par(mar=c(1,1,1,1))
image(seq(dim(volcano)[1]), seq(dim(volcano)[2]), volcano,
col=pal.2(length(breaks)-1), breaks=breaks-1e-8, xaxt="n", yaxt="n", ylab="", xlab="")
#Add additional graphics
levs=breaks
contour(seq(dim(volcano)[1]), seq(dim(volcano)[2]), volcano, levels=levs, add=TRUE)
#Add scale
par(mar=c(1,0,1,3))
imageScale(volcano, col=pal.2(length(breaks)-1), breaks=breaks-1e-8,
axis.pos=4, add.axis=FALSE)
axis(4,at=breaks, las=2)
box()
abline(h=levs)
par(op)
```

`isolines` *Interpolate isolines from ocean data*

Description

User defined isolines are interpolated from ocean data stored in a satin object and returned as polylines.

Usage

```
isolines(X, levels, period = 1, depth = 1, plot = TRUE)
```

Arguments

`X`	a satin object as returned by `read.nasaoc`, `read.ghrsst`, `read.osunpp`, or `read.cmems`.
`levels`	a vector of desired isolines in data units.
`period`	an integer referring to the image number to process.
`depth`	an integer referring to a depth level. Only for Copernicus data that includes variables at different depths.
`plot`	logical, if `TRUE` a plot is produced (the default).

Details

Polyline coordinates of desired isolines are calculated via `contourLines` and `convCP` (from **PBSMapping** package). Perhaps only interesting with sea surface temperatures (isotherms). For smoother isolines, the function `pixelate` can be used to spatialy re-scale the data. In order to use this function with Copernicus data returned from `read.cmems`, the appropriate indexing should be previously done.

Value

A list with two data frames of class `"PolySet"` and `"PolyData"`. See `help(PolySet)` and `help(PolyData)` for a complete description.

Author(s)

Héctor Villalobos, from code in examples of **PBSMapping** package

References

Schnute, J. T., Boers, N. M., Haigh, R. and Couture-Beil, A. 2011 PBS Mapping 2.62: user's guide revised from Canadian Technical Report of Fisheries and Aquatic Sciences 2549:vi + 112 p. Last updated Feb 29, 2011.

See Also

contourLines, convCP, PolySet

Examples

```
library(PBSmapping)
data(dsst)

# spatial re-scaling of the sst data for smoother isotherms
sst <- pixelate(dsst, extent = 0.2)
isotherms <- isolines(sst, levels = c(13, 17, 21))
plot(dsst, colbar = FALSE)
addLines(isotherms$PolySet, col = "black", lwd = 1.5)
```

---

pixelate *Spatial re-scaling of ocean data*

Description

This function down-scales ocean data by quadrants of user defined size (in degrees) according to a given function.

Usage

```
pixelate(X, extent = 0.25, FUN = mean)
```

Arguments

X	a satin object as returned by read.nasaoc, read.ghrsst, read.osunpp, or read.cmems.
extent	size in degrees of the squared quadrant for the new spatial scale.
FUN	function to be applied for obtaining the quadrants' values, defaults to mean.

Details

The main interest of this function is to obtain smooth isolines for ocean data (e.g. isotherms). In order to use this function with Copernicus data returned from read.cmems, the appropriate indexing should be done.

Value

An object of class "satin". See satin-class for details.

Author(s)

Héctor Villalobos

See Also

isolines

Examples

```
# load and plot sample SST data
data(dsst)
plot(dsst)

# change spatial resolution to 0.5 degrees
sst0.5 <- pixelate(dsst, extent = 0.5)
plot(sst0.5)
```

---

plot.satin *Visualise ocean data*

Description

Visual representation of ocean data from satellite images (MODIS, AVHRR, etc.) or derived from models (ocean productivity; CMEMS).

Usage

```
## S3 method for class 'satin'
plot(x, period = 1, depth = 1, xlim = NULL, ylim = NULL, zlim = NULL,
         map = NULL, map.col = "grey", map.outline = "black", scheme = "default",
           col.sep = 0.1, colbar = TRUE, main = NULL, main.pos = "topright",
         log = FALSE, units = NULL,  xaxt = "s", yaxt = "s", atx = NULL, aty = NULL,
   restore.par = TRUE, ...)
```

Arguments

x	a satin object as returned by read.nasaoc, read.ghrsst, read.osunpp, or read.cmems.
period	an integer referring to the image number to plot.
depth	an integer referring to a depth level. Only for Copernicus data that includes variables at different depths.
xlim	minimum and maximum longitude values for the map.
ylim	minimum and maximum latitude values for the map.
zlim	minimum and maximum ocean data values.
map	a user defined map of class "SpatialPolygons".
map.col	color for the map.
map.outline	color for the map outline.
scheme	color scheme for the ocean data, either: "default" or a vector of valid color names.

col.sep	separation between colors in data units.
colbar	logical. If TRUE a color bar is added to the map.
main	title for the map, defaults to the corresponding selected image period$tmStart in x.
main.pos	position for the title: "topright", "topleft", "bottomright" or "bottomleft".
log	logical. If TRUE, the ocean variable is log transformed (base 10). This is useful for chlorophyll concentration data.
units	color bar label, defaults to units in x.
xaxt, yaxt	"n" to supress axis.
atx, aty	specify tick-mark locations for fine tuning axes annotations.
restore.par	logical. If TRUE (the default), the graphical device is restored to its original state.
...	further arguments to pass to plot.

Details

The only mandatory argument for this function to produce a map is an object of class "satin" (x in this case). All the other arguments are optional and can be used to customize the plot. For chlorophyll concentration data, when log = TRUE, the color scale is fixed to a maximum value of 20, which implies that values above it are set to this maximum before applying logarithms. Using restore.par = FALSE to annotate plot, e.g. add isolines.

Value

A plot including a map, of the corresponding parameter is produced.

Author(s)

Héctor Villalobos

See Also

read.nasaoc, read.ghrsst, read.osunpp, and read.cmems

Examples

```
# SST data (Aqua Modis)
library(sp)
data(dsst, dmap) #load sample data and map

# default plot
plot(dsst)

# adding a custom map
plot(dsst, map = dmap)

# Chl-a concentration data (Aqua Modis) in actual units and in logarithmic scale
data(dchl)
plot(dchl, map = dmap, xlim = c(-120, -105), ylim = c(20, 35))
```

```
dev.new()
plot(dchl, map = dmap, xlim = c(-120, -105), ylim = c(20, 35), log = TRUE)

# Copernicus data
data(dcmems)
plot(dcmems$thetao, map = dmap, period = 2, depth = 5)
```

---

quiver *Vectors of ocean currents velocities*

Description

Plot vectors of ocean current velocities of varying size or color for Copernicus data.

Usage

```
quiver(u, v, period = 1, depth = 1, xlim = NULL, ylim = NULL, scale = 1,
       length = 0.05, colarrow = NULL, scheme = "default", ra.pos = NULL,
       ra.speed = NULL, map = NULL, map.col = "grey", map.outline = "black",
       colbar = FALSE, main = NULL, main.pos = "topright", add2map = FALSE, ...)
```

Arguments

u	a satin object with eastward ocean current velocities (in m/s).
v	a satin object with northward ocean current velocities (in m/s).
period	an integer referring to the image number to plot.
depth	an integer referring to a depth level.
xlim	minimum and maximum longitude values for the map.
ylim	minimum and maximum latitude values for the map.
scale	this argument controls the arrow length.
length	this value controls the size of the arrow head.
colarrow	color(s) for the arrows; see details.
scheme	color scheme for the arrows. Either: "default" or a vector of valid color names.
ra.pos	position (lon, lat) for a reference arrow.
ra.speed	speed (in m/s) of the reference arrow. The mean is used by default.
map	a user defined map of class "SpatialPolygons".
map.col	color for the map.
map.outline	color for the map outline.
colbar	logical. If TRUE a color bar is added to the plot.
main	title for the map, defaults to the corresponding selected image period.
main.pos	position for the title: "topright", "topleft", "bottomright" or "bottomleft".
add2map	a logical value. If FALSE an entire new plot is created. If TRUE ocean current velocity arrows are added to an existing plot.
...	further arguments to pass to arrows.

Details

Usage and behavior of this function closely resembles that of `plot.satin`. It differs in that two different satin objects are needed, one for each component of ocean current velocities. Also, the ocean current vectors can be overlayed to an existing plot of another variable, as the temperature, for instance.

Value

A map of the ocean current velocities is produced.

Author(s)

Héctor Villalobos

Examples

```
if(interactive()){
# import copernicus data with potential temperatures and ocean current velocities
oc <- read.cmems("global-reanalysis-phy-001-030-daily_1560792767602_GC_2013-14.nc")

# Default plot
quiver(u = oc$uo, v = oc$vo)

# Using default colors for the arrows, color bar and reference arrow
quiver(u = oc$uo, v = oc$vo, colarrow = TRUE, colbar = TRUE,
       ra.pos = c(-108, 30))

# Adding custom map and colors, and plotting another period and depth level
quiver(u = oc$uo, v = oc$vo, image = 4, depth = 10, colbar = TRUE,
       colarrow = TRUE, scheme = c("mediumblue", "lightcyan", "red"),
       map = dmap, ra.pos = c(-108, 30), ra.speed = 0.2)


# Overlay arrows to an existing plot of the potential temperature
plot(oc$thetao, map = dmap, xlim = c(-112, -108), ylim = c(24, 28),
     colbar = FALSE)
quiver(u = oc$uo, v = oc$vo, colarrow = TRUE, ra.pos = c(-108.5, 27),
       ra.speed = 0.15, main = "", length=0.02, scale = 0.7, add2map = TRUE)
}
```

`read.cmems` *Read ocean products data from CMEMS*

Description

Read data from netCDF files downloaded from Copernicus Marine Service (CMEMS).

Usage

```
read.cmems(nc)
```

Arguments

nc	name of the netCDF file to be read.

Details

The function can read data from netCDF files downloaded from CMEMS web page (https://marine.copernicus.eu/). It has been tested with Global Ocean Physical Reanalysis, and Global Ocean Biogeochemistry Hindcast databases. When downloading Copernicus data files, the user can select one or several variables (see references below), depth levels, daily or monthly means, time and spatial coverage. This makes every Copernicus netCDF file unique. The `read.cmems` function attempts to read all the variables and depth levels at once. When only one variable is included in the file, a `"satin"` object with one additional dimension in the data array to accommodate the different depth levels, is produced. On the other hand, when several variables are present, the function returns a named list, with every element being a `"satin"` object corresponding to each variable.

Value

A named list, with elements corresponding to each variable included in the netCDF file. Each list element (or when only one variable is present) is an object of class `"satin"`. See `satin-class` for details.

Author(s)

Héctor Villalobos

References

https://resources.marine.copernicus.eu/?option=com_csw&task=results

Fernandez, E. and Lellouche, J. M. 2018 Product User Manual for the Global Ocean Physical Reanalysis product, issue 1.1. EU Copernicus Marine Service.

Garric, G. and Parent, L. 2018 Product User Manual for the Global Ocean Reanalysis Products, issue 3.4. EU Copernicus Marine Service.

Perruche, C. 2019 Product User Manual for the Global Ocean Biogeochemistry Hindcast, issue 1.1. EU Copernicus Marine Service.

See Also

`read.nasaoc` for reading NASA's oceancolor web page data, `read.ghrsst` for JPL MUR SST data, and `read.osunpp` for reading ocean productivity data.

Examples

```
if(interactive()){
 # read Copernicus netcdf file with one variable
 cmems <- read.cmems("global-reanalysis-phy-001-030-monthly_1553098099635.nc")
 cmems
 str(cmems)

 # plotting the first period
```

```
plot(cmems)

# plotting the second period
plot(cmems, period = 2)

# read Copernicus netCDF file with several variables and depth levels
cmems2 <- read.cmems("global-reanalysis-phy-001-030-monthly_1553051256930.nc")
cmems2 # it is a list

# readed variables
names(cmems2)

# example plots for potential temperature
thetao <- cmems2$thetao
thetao
thetao$depth # depth levels
plot(thetao) # potential temperature at first depth level (0.49 m) and first month
plot(thetao, period = 2, depth = 5) # for fifth depth level (5.078 m), second month
}
```

`read.ghrsst` *Read sea surface temperature data from JPL Multi-scale Ultra-high Resolution (MUR) SST Project*

Description

Read, for a user defined area, SST data from netCDF files downloaded from JPL's Physical Oceanography Distributed Active Archive Center (podaac) web page.

Usage

```
read.ghrsst(nc, lons, lats)
```

Arguments

`nc`	name of the netCDF file(s) to be read.
`lons`	a vector with western- and eastern-most longitudes.
`lats`	a vector with southern- and northern-most latitudes.

Details

This function can read SST data from one or multiple netCDF files downloaded from Caltech Jet Propulsion Laboratory web page (https://podaac-opendap.jpl.nasa.gov/opendap/allData/ghrsst/data/GDS2/L4/GLOB/JPL/MUR/v4.1/). The user must specify the minimum and maximum latitude and longitude limits within the interval (-90, 90) and (-180, 180) respectively.

Value

An object of class `"satin"`. See `satin-class` for details.

Author(s)

Héctor Villalobos and Eduardo González-Rodríguez

References

https://www.ghrsst.org/

See Also

read.nasaoc for reading NASA's oceancolor web page data, read.osunpp for reading ocean productivity data, and read.cmems for data from Copernicus Marine Service.

Examples

```
if(interactive()){
 # read a single file, look at its structure and plot
 sst1km <- read.nasaoc("20130101090000-JPL-L4_GHRSST-SSTfnd-MUR-GLOB-v02.0-fv04.1.nc",
                       lats=c(20, 30), lons=c(-130, -105))
 sst1km
 str(sst1km)
 plot(sst1km)

 # read several files residing in the working directory
 files <- list.files(pattern = glob2rx("*.nc"))
 lats <- c(20, 30)
 lons <- c(-130, -105)
 mSST <- read.ghrsst(files, lats, lons)

 # plotting the first processed file
 plot(mSST)

 # plotting the second processed file
 plot(mSST, period = 2)
}
```

---

read.nasaoc *Read ocean data from NASA's Oceancolor web files*

Description

Read, for a user defined area, data from netCDF files downloaded from NASA's Oceancolor web page.

Usage

```
read.nasaoc(nc, lons, lats)
```

Arguments

nc	name of the netCDF file(s) to be read.
lons	a vector with western- and eastern-most longitudes.
lats	a vector with southern- and northern-most latitudes.

Details

This function can read data from one or multiple netCDF files downloaded from NASA's oceancolor web page (https://oceancolor.gsfc.nasa.gov/l3/). It has been tested mainly with sea surface temperature and chlorophyll concentration data from MODIS-Aqua and SeaWiFS sensors. It also works with Pathfinder AVHRR files v 5.2 and 5.3 (https://www.ncei.noaa.gov/products/avhrr-pathfinder-sst). The user must specify the minimum and maximum longitude and latitude limits within the interval (-180, 180) and (-90, 90) respectively.

Value

An object of class "satin". See satin-class for details.

Author(s)

Héctor Villalobos and Eduardo González-Rodríguez

See Also

read.ghrsst for JPL MUR SST data, read.osunpp for reading ocean productivity data, and read.cmems for data from Copernicus Marine Service.

Examples

```
if(interactive()){
 # read a single file, look at it and plot
 sst <- read.nasaoc("A20130892013096.L3m_8D_SST4_sst4_4km.nc",
                        lats=c(20, 30), lons=c(-130, -105))
 sst
 str(sst)
 plot(sst)

 # read several files residing in the working directory
 files <- list.files(pattern = glob2rx("*.nc"))
 lats <- c(20, 30)
 lons <- c(-130, -105)
 mSST <- read.nasaoc(files, lats, lons)

 # plotting the first processed file
 plot(mSST)

 # plotting the second processed file
 plot(mSST, period = 2)
}
```

read.osunpp *Read ocean productivity data from Oregon State University files*

Description

Read, for a user defined area, net primary production data from hdf files downloaded from OSU's Ocean Productivity home page.

Usage

```
read.osunpp(h5, lons, lats)
```

Arguments

h5	name of the hdf file(s) to be read.
lons	a vector with western- and eastern-most longitudes.
lats	a vector with southern- and northern-most latitudes.

Details

This function reads Oregon State University ocean productivity data files (http://sites.science.oregonstate.edu/ocean.productivity/standard.product.php). However, downloaded files should be first converted from hdf version 4 to version 5 with h4h5tools (https://portal.hdfgroup.org/display/support/Download+h4h5tools). The user must specify the minimum and maximum latitude and longitude limits within the interval (-90, 90) and (-180, 180) respectively.

Value

An object of class "satin". See satin-class for details.

Author(s)

Héctor Villalobos and Eduardo González-Rodríguez

See Also

read.nasaoc for reading NASA's oceancolor web page data, read.ghrsst for JPL MUR SST data, and read.cmems for data from Copernicus Marine Service.

Examples

```
if(interactive()){
 # read a single file, look at its structure and plot
 npp <- read.osunpp("vgpm.2013089.h5",
                       lats=c(20, 30), lons=c(-130, -105))
 npp
 str(npp)
 plot(npp)
```

```
 # read several files residing in the working directory
 files <- list.files(pattern = glob2rx("*.h5"))
 lats <- c(20, 30)
 lons <- c(-130, -105)
 mNPP <- read.osunpp(files, lats, lons)

 # plotting the first processed file
 plot(mNPP)

 # plotting the second processed file
 plot(mNPP, period = 2)
}
```

`satin-class` *Class "satin"*

Description

class for ocean data derived from satellites, imported from netCDF files.

Objects from the Class

Objects of class `"satin"` are created by importing functions: `read.nasaoc`, `read.ghrsst`, `read.cmems`, and `read.osunpp`. They can also be created, though not recommended, by direct calls to `satin`.

Slots

lon Object of class `"numeric"`; vector of longitudes of the area of interest (aoi).

lat Object of class `"numeric"`; vector of latitudes of the aoi.

data Object of class `"array"`; three or four dimensions array of ocean data corresponding to the lengths of `lat`, `lon`, `period$tmStart`, and `depth` vectors. The fourth dimension associated to depth levels will only be present in `"satin"` objects derived from Copernicus data files.

attribs Object of class `"list"`; attributes of the data stored in the netCDF file: `title`: title attribute from netCDF file; `longname`: long name of ocean data variable; `name`: short name of ocean data variable; `units`: ocean data variable measurement units; `temporal_range`: averaging period length, i.e. daily, monthly, etc.; `spatial_resolution`: spatial resolution of the ocean data variable.

period Object of class `"list"`; a vector with start time (`tmStart`) and a vector with end time (`tmEnd`) of the averaging period of the data. Both vectors are of class `"POSIXct"`.

depth Object of class `"numeric"`; vector of depth levels in the data. Only meaningful for `"satin"` objects derived from Copernicus data files.

Methods

plot `signature(x = "satin", y = "missing")`: plot method for `"satin"` objects. See details in `plot.satin`

show `signature(object = "satin")`: display a summary of the object. See example below

Author(s)

Héctor Villalobos

See Also

See `read.nasaoc`; `read.osunpp`; `read.ghrsst`; and `read.cmems` for functions producing `"satin"` objects, and `plot.satin` for a complete description of plotting arguments.

Examples

```
data(dsst)
class(dsst)
dsst
str(dsst)
```

`satinDataframe` *Reshape a satin object into a data frame*

Description

A satin object is reshaped as a data frame with longitude, latitude and remote sensed data column(s).

Usage

```
satinDataframe(X, reverse = FALSE)
```

Arguments

`X`	a satin object as returned by `read.nasaoc`, `read.ghrsst`, `read.osunpp`, `read.cmems`, or a data frame produced with this function.
`reverse`	logical, `TRUE` is used to create a satin object from a data frame previously created with this function (see details).

Details

This is mainly an utility function called by `crop`, `extractPts`, and `pixelate` functions. However, it may be useful when the user wants to arrange the ocean data variables stored in a satin object as a data frame, with rows representing pixels whose longitude and latitude coordinates are in the first two columns and their data values in the remaining column(s). The attributes describing the data (name, units, period, etc.) are preserved. Specifying `reverse = TRUE` reverses the operation.

Value

When `reverse = FALSE`, a data frame with at least three columns: "x", "y" for longitude and latitude coordinates, and data for the variable(s). When `reverse = TRUE` a satin object as produced.

Author(s)

Héctor Villalobos

Examples

```
# load sample SST data
data(dsst)

# reshape into a data frame
X <- satinDataframe(dsst)
head(X)

# reverse the operation
Y <- satinDataframe(X, reverse = TRUE)
Y
plot(Y)
```

---

satinMean *Averaging ocean data by time period*

Description

This function allows to average ocean data by a selected time period. For example, from daily data, the user may want to obtain weekly, monthly, quarterly or even yearly averages.

Usage

```
satinMean(X, by = "%m", depth = NULL)
```

Arguments

X	a satin object as returned by read.nasaoc, read.ghrsst, read.osunpp, or read.cmems.
by	a character string specifying the averaging time period, see details.
depth	an integer referring to a given depth level (defaults to 1). Only meaningful for Copernicus data that includes variables at different depths.

Details

Accepted strings in by follow the notation of conversion specifications described in strptime. Currently implemented: "%Y", "%m", "%Y-%m", "%j", "%qtr", "%Y-%qtr", "%sem", "%Y-%sem", "%U", "%V", "%W", "%Y-%U", "%Y-%V", "%Y-%W". This includes two non-standard strings and their combinations with year: "%qtr", and "%sem", for quarter and semester, respectively, which are passed to lubridate's functions of the same names. This function is meant to be used in combination with anomaly to obtain anomalies with respect to an average period.

Value

An object of class "satin" (see satin-class for details) where the third dimension in the data array accomodates averages by time period. An extra element (labels) is included in the slot attribs to identify the time period used for the averages.

Author(s)

Héctor Villalobos and Eduardo González-Rodríguez

See Also

anomaly for calculating anomalies of ocean data, and climatology for climatologies.

Examples

```
if(interactive()){
sst.ym <- satinMean(sst, by = "%Y-%m") # sst is a satin object with 240 weekly
                                       # images, totalling five years of data.

sst.ym # this will have 60 time periods, one for every month and year.

sst.m <- satinMean(sst, by = "%m") # this will produce 12 monthly periods
}
```

---

satinPalette *Define custom color palettes for echograms*

Description

This function allows to design and visualise color palettes to be used in ocean data plots.

Usage

```
satinPalette(zmin, zmax, col.sep = 0.1, scheme = "default", visu = FALSE)
```

Arguments

zmin	lower limit in data units.
zmax	upper limit in data units.
col.sep	separation between colors in data units.
scheme	color scheme, either: "default" or a vector of valid color names.
visu	logical. If TRUE, a visual representation of the palette is displayed.

Details

This function is mainly intended to be called by plot.satin, however it is possible to use it directly in order to have a first impression of a custom color palette.

Value

A list with two elements

palette	a vector of colors.
breaks	a vector of color breaks.

Author(s)

Héctor Villalobos

See Also

plot.satin

Examples

```
satinPalette(zmin = 10, zmax = 35, visu = TRUE)
satinPalette(zmin = 10, zmax = 35, col.sep = 0.2, scheme = c("white", "blue", "black"), visu = TRUE)
```

---

velocity *Ocean currents velocities*

Description

Calculate ocean current speed and direction from eastward and northward components (Copernicus data).

Usage

```
velocity(u, v, depth = 1)
```

Arguments

u	a satin object with eastward ocean current velocities (in m/s).
v	a satin object with northward ocean current velocities (in m/s).
depth	an integer referring to a depth level.

Details

This function calculates the resulting velocity vectors in the same way as the "quiver" function. In addition, these values are returned togheter with their direction in degrees.

Value

A list with 2 elements, "speed" and "rhumb", both of class "satin". The former contains the resulting velocity vectors as drawn by "quiver", while the second their direction in degrees.

Author(s)

Héctor Villalobos

Examples

```
if(interactive()){
# import copernicus data with potential temperatures and ocean current velocities
oc <- read.cmems("global-reanalysis-phy-001-030-daily_1560792767602_GC_2013-14.nc")

vel <- velocity(u = oc$uo, v = oc$vo)

# Plots overlaying velocity vectors
plot(vel$speed)
quiver(u = oc$uo, v = oc$vo, main = "", length=0.02, scale = 0.7, add2map = TRUE)

plot(vel$rhumb, col.sep = 90, scheme = c('#d7301f', '#fc8d59','#fdcc8a', '#fef0d9'))
quiver(u = oc$uo, v = oc$vo, main = "", add2map = TRUE)
}
```

---

verticalProfiles *CTD-like vertical profiles for Copernicus variables*

Description

Extract ocean data available at different depth levels (Copernicus data) and plot CTD-like vertical profiles.

Usage

```
verticalProfiles(X, point = NULL, xlim = NULL, ylim = NULL)
```

Arguments

X	a satin object with Copernicus data registered at different depths (i.e. potential temperature (thetao), salinity (so), etc.).
point	a data frame with point longitude and latitude coordinates where to extract data values.
xlim	limits for data variable.
ylim	vertical depth limits.

Details

Selected Copernicus data should be available at several depths, otherwise is meaningless. Also, if more than one point is selected, only the first will be processed.

Author(s)

Héctor Villalobos

Examples

```
# load sample Copernicus data
data(dcmems)
# available variables
names(dcmems)

# plot potential temperature
plot(dcmems$thetao)

# define point for extracting and plotting vertical profile
pt <- data.frame(lon = -108.5, lat = 23.33)
vp <- verticalProfiles(dcmems$thetao, point = pt)
# inspect extracted data
vp
```

Index