Package: climateR (via r-universe)

December 5, 2024

Type Package

Title climateR

Description Find, subset, and retrive geospatial data by AOI.

Version 0.3.7

Maintainer Mike Johnson <mike.johnson@noaa.gov>

BugReports https://github.com/mikejohnson51/climateR/issues

URL https://github.com/mikejohnson51/climateR

Depends R(>= 3.5.0)

Imports arrow, dplyr, future.apply, glue, gifski, ncmeta, RNetCDF, rnz, terra, utils, methods, stats, grDevices

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests httr, AOI, covr, testthat (>= 3.0.0), rmarkdown, knitr, kableExtra, distill, ggplot2, patchwork, sf, tidyr, tidyterra, reticulate

Remotes mikejohnson51/AOI, dblodgett-usgs/ncmeta, DOI-USGS/rnz

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libnetcdf-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://owp-spatial.r-universe.dev

RemoteUrl https://github.com/mikejohnson51/climateR

RemoteRef HEAD

RemoteSha 718a8d0357e9249894fb4f6e922911042db47e79

Contents

`.resource_grid` *Extract grid metadata from NC Pointer*

Description

Extract grid metadata from NC Pointer

Usage

```
.resource_grid(URL, X_name = NULL, Y_name = NULL, stopIfNotEqualSpaced = TRUE)
```

Arguments

`URL`	location of data to process
`X_name`	Name of X diminsion. If NULL it is found
`Y_name`	Name of Y diminsion. If NULL it is found
`stopIfNotEqualSpaced`	stop if not equal space grid

Value

list

See Also

Other dap: .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

.resource_grid_zarr *Extract Grid Information from Zarr Resource*

Description

Retrieves grid dimension metadata and calculates grid properties.

Usage

```
.resource_grid_zarr(
  URL,
  X_name = NULL,
  Y_name = NULL,
  stopIfNotEqualSpaced = TRUE
)
```

Arguments

URL	Character. The URL of the Zarr file.
X_name	Character. Name of the X-coordinate variable. Defaults to NULL.
Y_name	Character. Name of the Y-coordinate variable. Defaults to NULL.
stopIfNotEqualSpaced	Logical. Whether to stop if grid cells are not equally spaced. Defaults to TRUE.

Value

A data frame with grid properties.

See Also

Other zarr: .resource_time_zarr(), go_get_zarr(), read_zarr_file(), zarr_crop(), zarr_get(), zarr_to_terra(), zarr_xyzv()

.resource_time *Extract time metadata from NC Pointer*

Description

Extract time metadata from NC Pointer

Usage

```
.resource_time(URL, T_name = NULL)
```

Arguments

URL	location of data to process
T_name	Name of T dimension. If NULL it is found

Value

list

See Also

Other dap: .resource_grid(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

.resource_time_zarr *Extract Time Information from Zarr Resource*

Description

Retrieves time dimension metadata and calculates time intervals.

Usage

```
.resource_time_zarr(URL, T_name = NULL)
```

Arguments

URL	Character. The URL of the Zarr file.
T_name	Character. Name of the time variable. Defaults to NULL.

Value

A list with time duration, interval, and count information.

See Also

Other zarr: `.resource_grid_zarr()`, `go_get_zarr()`, `read_zarr_file()`, `zarr_crop()`, `zarr_get()`, `zarr_to_terra()`, `zarr_xyzv()`

`animation` *Animate Object as GIF*

Description

Animate a SpatRaster object as a gif.

Usage

```
animation(data, AOI = NULL, feild_pattern = NULL, outfile, colors = blues9)
```

Arguments

`data`	a SpatVect or sf object
`AOI`	optional AOI sf or SpatVect object to overlay on gif
`feild_pattern`	optional string vector to filter the desired attributes by
`outfile`	path to write gif file, must have .gif extenstion
`colors`	colors to plot with

Value

file.path

See Also

Other viz: `animation_raster()`, `animation_vector()`

animation_raster *Animate SpatRast as GIF*

Description

Animate a SpatRaster object as a gif.

Usage

```
animation_raster(data, AOI = NULL, outfile, colors = blues9)
```

Arguments

data	a single SpatRast object
AOI	optional AOI sf or SpatVect object to overlay on gif
outfile	path to write gif file, must have .gif extenstion
colors	colors to plot with

Value

file.path

See Also

Other viz: animation(), animation_vector()

animation_vector *Animate vector as GIF*

Description

Animate a sf or SpatVect object as a gif.

Usage

```
animation_vector(data, feild_pattern = NULL, outfile, colors = blues9)
```

Arguments

data	a SpatVect or sf object
feild_pattern	optional string vector to filter the desired attributes by
outfile	path to write gif file, must have .gif extenstion
colors	colors to plot with

Value

file.path

See Also

Other viz: `animation()`, `animation_raster()`

`catalog` *ClimateR Catalog*

Description

ClimateR Catalog

Usage

```
catalog
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 112398 rows and 29 columns.

See Also

Other catalog: `read_live_catalog()`

`checkDodsrc` *Check dodsrc file*

Description

Check that there is a netrc file with a valid entry for urs.earthdata.nasa.gov.

Usage

```
checkDodsrc(dodsrcFile = getDodsrcPath(), netrcFile = getNetrcPath())
```

Arguments

`dodsrcFile`	File path to dodsrc file to check.
`netrcFile`	File path to netrc file to check.

Value

logical

See Also

Other netrc: checkNetrc(), getDodsrcPath(), getNetrcPath(), writeDodsrc(), writeNetrc()

---

checkNetrc *Check netrc file*

Description

Check that there is a netrc file with a valid entry for urs.earthdata.nasa.gov.

Usage

```
checkNetrc(netrcFile = getNetrcPath(), machine = "urs.earthdata.nasa.gov")
```

Arguments

netrcFile	A character. File path to netrc file to check.
machine	the machine you are logging into

Value

logical

See Also

Other netrc: checkDodsrc(), getDodsrcPath(), getNetrcPath(), writeDodsrc(), writeNetrc()

---

climater_dap *ClimateR dry run*

Description

ClimateR dry run

Usage

```
climater_dap(id, args, verbose, dryrun, print.arg = FALSE)
```

Arguments

id	The resource name, agency, or catalog identifier
args	The parent function arguments
verbose	Should messages be emited?
dryrun	Return summary of data prior to retrieving it
print.arg	should arguments be printed? Usefull for debugging

Value

data.frame

See Also

Other dap: .resource_grid(), .resource_time(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

---

climater_filter *ClimateR Catalog Filter*

Description

Filter the climateR catalog based on a set of constraints

Usage

```
climater_filter(
  id = NULL,
  asset = NULL,
  AOI = NULL,
  startDate = NULL,
  endDate = NULL,
  varname = NULL,
  model = NULL,
  scenario = NULL,
  ensemble = NULL
)
```

Arguments

id	The resource, agency, or catalog identifier
asset	The subdataset or asset in a given resource
AOI	an sf of SpatVect point or polygon to extract data for
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
varname	variable name to extract (e.g. tmin)
model	GCM model name generating
scenario	A climate or modeling scenario
ensemble	The model ensemble member used to generate data

Value

data.frame

See Also

Other dap: `.resource_grid()`, `.resource_time()`, `climater_dap()`, `dap()`, `dap_crop()`, `dap_get()`, `dap_meta()`, `dap_summary()`, `dap_to_local()`, `dap_xyzv()`, `extract_sites()`, `get_data()`, `go_get_dap_data()`, `grid_meta()`, `make_ext()`, `make_vect()`, `merge_across_time()`, `parse_date()`, `read_dap_file()`, `read_ftp()`, `time_meta()`, `try_att()`, `var_to_terra()`, `variable_meta()`, `vrt_crop_get()`

---

dap *Get Data (Data Access Protocol)*

Description

this function provides a consistent data access protocol (dap) to a wide range of local and remote resources including VRT, TDS, NetCDF

Define and get data from a DAP resource

Usage

```
dap(
  URL = NULL,
  catalog = NULL,
  AOI = NULL,
  startDate = NULL,
  endDate = NULL,
  varname = NULL,
  grid = NULL,
  start = NULL,
  end = NULL,
  toptobottom = FALSE,
  ID = NULL,
  verbose = TRUE
)
```

Arguments

URL	local file path or URL
catalog	subset of open.dap catalog
AOI	an sf of SpatVect point or polygon to extract data for
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
varname	variable name to extract (e.g. tmin)

grid	a list containing an extent (), and crs
start	for non "dated" items, start can be called by index
end	for non "dated" items, end can be called by index
toptobottom	should data be inverse?
ID	a column of unique identifiers
verbose	Should dap_summary be printed?

Details

Wraps dap_get and dap_crop into one. If AOI is NULL no spatial crop is executed. If startDate AND endDate are NULL, no temporal crop is executed. If just endDate is NULL it defaults to the startDate.

Value

data.frame

See Also

Other dap: `.resource_grid()`, `.resource_time()`, `climater_dap()`, `climater_filter()`, `dap_crop()`, `dap_get()`, `dap_meta()`, `dap_summary()`, `dap_to_local()`, `dap_xyzv()`, `extract_sites()`, `get_data()`, `go_get_dap_data()`, `grid_meta()`, `make_ext()`, `make_vect()`, `merge_across_time()`, `parse_date()`, `read_dap_file()`, `read_ftp()`, `time_meta()`, `try_att()`, `var_to_terra()`, `variable_meta()`, `vrt_crop_get()`

---

dap_crop *Crop DAP file*

Description

Crop DAP file

Usage

```
dap_crop(
  URL = NULL,
  catalog = NULL,
  AOI = NULL,
  startDate = NULL,
  endDate = NULL,
  start = NULL,
  end = NULL,
  varname = NULL,
  verbose = TRUE
)
```

Arguments

URL	local file path or URL
catalog	subset of open.dap catalog
AOI	an sf of SpatVect point or polygon to extract data for
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
start	for non "dated" items, start can be called by index
end	for non "dated" items, end can be called by index
varname	variable name to extract (e.g. tmin)
verbose	Should dap_summary be printed?

Value

data.frame

See Also

Other dap: `.resource_grid()`, `.resource_time()`, `climater_dap()`, `climater_filter()`, `dap()`, `dap_get()`, `dap_meta()`, `dap_summary()`, `dap_to_local()`, `dap_xyzv()`, `extract_sites()`, `get_data()`, `go_get_dap_data()`, `grid_meta()`, `make_ext()`, `make_vect()`, `merge_across_time()`, `parse_date()`, `read_dap_file()`, `read_ftp()`, `time_meta()`, `try_att()`, `var_to_terra()`, `variable_meta()`, `vrt_crop_get()`

---

dap_get *Get DAP resource data*

Description

Get DAP resource data

Usage

```
dap_get(dap, varname = NULL)
```

Arguments

dap	data.frame from catalog or dap_crop
varname	name of variable to extract. If NULL, then get all

Value

SpatRaster

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

---

dap_meta *Find DAP Metadata*

Description

Find DAP Metadata

Usage

```
dap_meta(raw)
```

Arguments

raw	data.frame

Value

data.frame

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

---

dap_summary *Print Summary Information About a OpenDAP Resource*

Description

Print summary information about a DAP summary

Usage

```
dap_summary(dap = NULL, url = NULL)
```

Arguments

dap	data.frame from catalog or dap_crop
url	Unique Resource Identifier (http or local)

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

---

dap_to_local *Convert OpenDAP to start/count call*

Description

Convert OpenDAP to start/count call

Usage

```
dap_to_local(dap, get = TRUE)
```

Arguments

dap	dap description
get	should data be collected?

Value

numeric array

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

dap_xyzv *Get XYTV data from DAP URL*

Description

Get XYTV data from DAP URL

Usage

```
dap_xyzv(obj, varname = NULL, varmeta = FALSE)
```

Arguments

obj	an OpenDap URL or NetCDF object
varname	name of variable to extract. If NULL, then get all
varmeta	should variable metadata be appended?

Value

data.frame with (varname, X_name, Y_name, T_name)

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

extract_sites *Extract Sites*

Description

extract timeseries values from a raster stack for a set of points

Usage

```
extract_sites(r, pts, ID = NULL)
```

Arguments

r	a SpatRaster object
pts	point to extract from
ID	the unique identifier of each point (column name from pts)

Value

a data.frame with columes representing points, and rows time periods

See Also

Other dap: `.resource_grid()`, `.resource_time()`, `climater_dap()`, `climater_filter()`, `dap()`, `dap_crop()`, `dap_get()`, `dap_meta()`, `dap_summary()`, `dap_to_local()`, `dap_xyzv()`, `get_data()`, `go_get_dap_data()`, `grid_meta()`, `make_ext()`, `make_vect()`, `merge_across_time()`, `parse_date()`, `read_dap_file()`, `read_ftp()`, `time_meta()`, `try_att()`, `var_to_terra()`, `variable_meta()`, `vrt_crop_get()`

---

get3DEP *Get USGS 3DEP DEMs*

Description

Get USGS 3DEP DEMs

Usage

```
get3DEP(AOI, resolution = "30m", ID = NULL)
```

Arguments

`AOI`	an sf of SpatVect point or polygon to extract data for
`resolution`	DEM resolution (10m or 30m (default))
`ID`	a column of unique identifiers

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: `getBCCA()`, `getCABCM()`, `getCHIRPS()`, `getDaymet()`, `getGLDAS()`, `getGridMET()`, `getISRIC_soils()`, `getLCMAP()`, `getLOCA()`, `getLOCA_hydro()`, `getLivneh()`, `getLivneh_fluxes()`, `getMACA()`, `getMODIS()`, `getNASADEM()`, `getNLCD()`, `getNLDAS()`, `getPRISM()`, `getTerraClim()`, `getTerraClimNormals()`, `getVIC()`, `getWorldClim()`

getBCCA *Get BCCA data*

Description

Get BCCA data

Usage

```
getBCCA(
  AOI,
  varname,
  model = "CCSM4",
  scenario = "rcp45",
  ensemble = NULL,
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
model	GCM model name generating
scenario	A climate or modeling scenario
ensemble	The model ensemble member used to generate data
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

getCABCM *Get California Basin Characterization Model (CABCM) historical and projected climate and hydrology data.*

Description

The California Basin Characterization Model (CABCM) dataset provides historical and projected climate and hydrology data at a 270 meter resolution, which is relevant for watershed-scale evaluation and planning.

Usage

```
getCABCM(
  AOI = NULL,
  varname = NULL,
  model = "CNRM",
  scenario = NULL,
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
model	GCM model name generating
scenario	A climate or modeling scenario
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

getCHIRPS *Get CHIRPS data*

Description

Get CHIRPS data

Usage

```
getCHIRPS(
  AOI,
  varname = NULL,
  timeRes = "daily",
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
timeRes	"Pentad", "Annual", "Daily" (default), or "Monthly"
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

`getDaymet` *Get Daymet Climate Data for an Area of Interest*

Description

This dataset provides Daymet Version 4 model output data as gridded estimates of daily weather parameters for North America. Daymet output variables include the following parameters: minimum temperature, maximum temperature, precipitation, shortwave radiation, vapor pressure, snow water equivalent, and day length. The dataset covers the period from January 1, 1980 to December 31 of the most recent full calendar year. Each subsequent year is processed individually at the close of a calendar year after allowing adequate time for input weather station data to be of archive quality. Daymet variables are continuous surfaces provided as individual files, by year, at a 1-km x 1-km spatial resolution and a daily temporal resolution. Data are in a Lambert Conformal Conic projection for North America and are in a netCDF file format compliant with Climate and Forecast (CF) metadata conventions.

Usage

```
getDaymet(
  AOI,
  varname = NULL,
  startDate = NULL,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

`AOI`	an sf of SpatVect point or polygon to extract data for
`varname`	variable name to extract (e.g. tmin)
`startDate`	a start date given as "YYYY-MM-DD" to extract data for
`endDate`	an end date given as "YYYY-MM-DD" to extract data for
`verbose`	Should messages be emited?
`ID`	a column of unique identifiers
`dryrun`	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

---

getDodsrcPath *Get a default dodsrc file path*

Description

Get a default dodsrc file path

Usage

```
getDodsrcPath()
```

Value

A character vector containing the default netrc file path

See Also

Other netrc: checkDodsrc(), checkNetrc(), getNetrcPath(), writeDodsrc(), writeNetrc()

Examples

```
getDodsrcPath()
```

---

getGLDAS *Get GLDAS data*

Description

Get GLDAS data

Usage

```
getGLDAS(
  AOI,
  varname = NULL,
  model = NULL,
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
model	GCM model name generating
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

---

getGridMET *Get GridMet Climate Data for an Area of Interest*

Description

gridMET is a dataset of daily high-spatial resolution (~4-km, 1/24th degree) surface meteorological data covering the contiguous US from 1979-yesterday. These data are updated daily.

Usage

```
getGridMET(
  AOI,
  varname,
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

---

getISRIC_soils *Get World Soil Information gridded weather and climate data for historical (near current) conditions.*

Description

World Soil Information (International Soil Reference and Information Centre) serves the international community with open access global soil data

Usage

```
getISRIC_soils(AOI = NULL, varname = NULL, verbose = TRUE, ID = NULL)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
verbose	Should messages be emited?
ID	a column of unique identifiers

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

`getLCMAP` *Get USGS LCMAP*

Description

Land Change Monitoring, Assessment, and Projection

Usage

```
getLCMAP(AOI, year = 2019, type = "primary landcover", ID = NULL)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
year	Land cover product year 1985 - 2019 (default = 2019)
type	product type (primary landcover (default), secondary landcover, primary confidence, secondary confidence, cover change, change day, change magniture, model cquality, spectral stability, spectral lastchance)
ID	a column of unique identifiers

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

getLivneh *Get Livneh data*

Description

Get Livneh data

Usage

```
getLivneh(
  AOI,
  varname = NULL,
  startDate,
  endDate = NULL,
  timeRes = "daily",
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
timeRes	daily or monthly
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

getLivneh_fluxes *Get Livneh Flux data*

Description

Get Livneh Flux data

Usage

```
getLivneh_fluxes(
  AOI,
  varname = NULL,
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

getLOCA *Get LOCA Climate Data for an Area of Interest*

Description

LOCA is a statistical downscaling technique that uses past history to add improved fine-scale detail to global climate models. LOCA has been used to downscale 32 global climate models from the CMIP5 archive at a 1/16th degree spatial resolution, covering North America from central Mexico through Southern Canada. The historical period is 1950-2005, and there are two future scenarios available: RCP 4.5 and RCP 8.5 over the period 2006-2100 (although some models stop in 2099). The variables currently available are daily minimum and maximum temperature, and daily precipitation. For more information visit: http://loca.ucsd.edu/.

Usage

```
getLOCA(
  AOI,
  varname,
  model = "CCSM4",
  scenario = "rcp45",
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
model	GCM model name generating
scenario	A climate or modeling scenario
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

`getLOCA_hydro` *Get LOCA Hydrology data*

Description

Get LOCA Hydrology data

Usage

```
getLOCA_hydro(
  AOI,
  varname,
  model = "CCSM4",
  scenario = "rcp45",
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
model	GCM model name generating
scenario	A climate or modeling scenario
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

`getMACA` *Get MACA Climate Data for an Area of Interest*

Description

Multivariate Adaptive Constructed Analogs (MACA) is a statistical method for downscaling Global Climate Models (GCMs) from their native coarse resolution to a higher spatial resolution that captures reflects observed patterns of daily near-surface meteorology and simulated changes in GCMs experiments.

Usage

```
getMACA(
  AOI,
  varname,
  timeRes = "day",
  model = "CCSM4",
  scenario = "rcp45",
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
timeRes	daily or monthly
model	GCM model name generating
scenario	A climate or modeling scenario
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

---

getMODIS *Get MODIS data*

Description

Get MODIS data

Usage

```
getMODIS(
  AOI,
  asset = NULL,
  varname = NULL,
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
asset	The MODIS sensor
varname	variable name to extract (e.g. tmin)
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

---

getNASADEM *Get NASA Global DEM*

Description

Get NASA Global DEM

Usage

```
getNASADEM(AOI, ID = NULL)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
ID	a column of unique identifiers

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

---

getNetrcPath *Get the default netrc file path*

Description

Get a default netrc file path

Usage

```
getNetrcPath()
```

Value

A character vector containing the default netrc file path

See Also

Other netrc: checkDodsrc(), checkNetrc(), getDodsrcPath(), writeDodsrc(), writeNetrc()

Examples

```
getNetrcPath()
```

`getNLCD` *Get USGS National Land Cover Dataset*

Description

Get USGS National Land Cover Dataset

Usage

```
getNLCD(AOI, year = 2019, type = "land cover", ID = NULL)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
year	Landcover product year (2001, 2011,2016,2019)
type	product type
ID	a column of unique identifiers

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

getNLDAS *Get NLDAS data*

Description

Get NLDAS data

Usage

```
getNLDAS(
  AOI,
  varname = NULL,
  model = "FORA0125_H.002",
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
model	GCM model name generating
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

getPRISM *Get PRISM data*

Description

Get PRISM data

Usage

```
getPRISM(
  AOI,
  varname = NULL,
  startDate,
  endDate = NULL,
  timeRes = "daily",
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
timeRes	daily or monthly
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getTerraClim(), getTerraClimNormals(), getVIC(), getWorldClim()

getTerraClim *Get Terra Climate Data for an Area of Interest*

Description

Get Terra Climate Data for an Area of Interest

Usage

```
getTerraClim(
  AOI,
  varname = NULL,
  startDate = NULL,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClimNormals(), getVIC(), getWorldClim()

`getTerraClimNormals` *Get TerraClimate Normals for an Area of Interest*

Description

These layers from TerraClimate were creating using climatically aided interpolation of monthly anomalies from the CRU Ts4.0 and Japanese 55-year Reanalysis (JRA-55) datasets with WorldClim v2.0 climatologies.

Usage

```
getTerraClimNormals(
  AOI,
  varname,
  scenario = "19812010",
  month = 1:12,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

`AOI`	an sf of SpatVect point or polygon to extract data for
`varname`	variable name to extract (e.g. tmin)
`scenario`	A climate or modeling scenario
`month`	numeric. and month or vector of months to access. Default is 1:12
`verbose`	Should messages be emited?
`ID`	a column of unique identifiers
`dryrun`	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: `get3DEP()`, `getBCCA()`, `getCABCM()`, `getCHIRPS()`, `getDaymet()`, `getGLDAS()`, `getGridMET()`, `getISRIC_soils()`, `getLCMAP()`, `getLOCA()`, `getLOCA_hydro()`, `getLivneh()`, `getLivneh_fluxes()`, `getMACA()`, `getMODIS()`, `getNASADEM()`, `getNLCD()`, `getNLDAS()`, `getPRISM()`, `getTerraClim()`, `getVIC()`, `getWorldClim()`

getVIC *Get VIC data*

Description

Get VIC data

Usage

```
getVIC(
  AOI,
  varname,
  model = "CCSM4",
  scenario = "rcp45",
  startDate,
  endDate = NULL,
  verbose = FALSE,
  ID = NULL,
  dryrun = FALSE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
model	GCM model name generating
scenario	A climate or modeling scenario
startDate	a start date given as "YYYY-MM-DD" to extract data for
endDate	an end date given as "YYYY-MM-DD" to extract data for
verbose	Should messages be emited?
ID	a column of unique identifiers
dryrun	Return summary of data prior to retrieving it

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getWorldClim()

getWorldClim *Get WorlClim gridded weather and climate data for historical (near current) conditions.*

Description

WorldClim is a database of high spatial resolution global weather and climate data. These data can be used for mapping and spatial modeling.

Usage

```
getWorldClim(
  AOI = NULL,
  varname = NULL,
  model = "wc2.1_5m",
  month = 1:12,
  ID = NULL,
  verbose = TRUE
)
```

Arguments

AOI	an sf of SpatVect point or polygon to extract data for
varname	variable name to extract (e.g. tmin)
model	GCM model name generating
month	numeric. and month or vector of months to access. Default is 1:12
ID	a column of unique identifiers
verbose	Should messages be emited?

Value

if AOI is polygon a list of SpatRasters, if AOI is a point then a data.frame of modeled records.

See Also

Other shortcuts: get3DEP(), getBCCA(), getCABCM(), getCHIRPS(), getDaymet(), getGLDAS(), getGridMET(), getISRIC_soils(), getLCMAP(), getLOCA(), getLOCA_hydro(), getLivneh(), getLivneh_fluxes(), getMACA(), getMODIS(), getNASADEM(), getNLCD(), getNLDAS(), getPRISM(), getTerraClim(), getTerraClimNormals(), getVIC()

get_data *Get DAP Array*

Description

Get DAP Array

Usage

```
get_data(dap)
```

Arguments

dap	dap description

Value

SpatRast

See Also

Other dap: `.resource_grid()`, `.resource_time()`, `climater_dap()`, `climater_filter()`, `dap()`, `dap_crop()`, `dap_get()`, `dap_meta()`, `dap_summary()`, `dap_to_local()`, `dap_xyzv()`, `extract_sites()`, `go_get_dap_data()`, `grid_meta()`, `make_ext()`, `make_vect()`, `merge_across_time()`, `parse_date()`, `read_dap_file()`, `read_ftp()`, `time_meta()`, `try_att()`, `var_to_terra()`, `variable_meta()`, `vrt_crop_get()`

go_get_dap_data *Read formated DAP URL as SpatRast*

Description

Read formated DAP URL as SpatRast

Usage

```
go_get_dap_data(dap)
```

Arguments

dap	output from dap_crop

Value

SpatRast

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

`go_get_zarr` *Retrieve data from a Zarr resource*

Description

This function retrieves or prepares metadata for data stored in a Zarr format.

Usage

```
go_get_zarr(zarr, get = TRUE)
```

Arguments

`zarr`	A data frame containing details of the Zarr resource to process. Each row should correspond to a single Zarr resource.
`get`	Logical. If 'TRUE', retrieves data; if 'FALSE', returns metadata information.

Value

If 'get = TRUE', returns the requested data as a matrix. If 'get = FALSE', returns a data frame containing metadata information for the resource.

See Also

Other zarr: .resource_grid_zarr(), .resource_time_zarr(), read_zarr_file(), zarr_crop(), zarr_get(), zarr_to_terra(), zarr_xyzv()

Examples

```
## Not run:
# Example usage (assuming `zarr` is a properly formatted data frame):
# result <- go_get_zarr(zarr, get = TRUE)

## End(Not run)
```

grid_meta *Find DAP grid metadata*

Description

Find DAP grid metadata

Usage

```
grid_meta(raw)
```

Arguments

raw	data.frame

Value

data.frame

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

make_ext *Convert catalog entry to extent*

Description

Convert catalog entry to extent

Usage

```
make_ext(cat)
```

Arguments

cat	catalog entry (data.frame with an (Xn, X1, Yn, Y1, crs)

Value

SpatExtent

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

---

make_vect *Make Vector*

Description

Make Vector

Usage

```
make_vect(cat)
```

Arguments

cat	catalog entry (data.frame with an c(Xn, X1, Yn, Y1, crs))

Value

SpatVect

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

---

merge_across_time *Merge List of SpatRaster's across time*

Description

Given a list of SpatRasters with possibly shared names, merge across time

Usage

```
merge_across_time(data)
```

Arguments

data	list of names SpatRasters

Value

data.frame with (varname, X_name, Y_name, T_name)

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

---

parse_date *Parse Dates from duration and interval*

Description

Parse Dates from duration and interval

Usage

```
parse_date(duration, interval)
```

Arguments

duration	time duration
interval	time interval

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

read_dap_file *Read from a OpenDAP landing page*

Description

Reads an OpenDap resources and returns metadata

Usage

```
read_dap_file(URL, varname = NULL, id, varmeta = TRUE)
```

Arguments

URL	URL to OpenDap resource
varname	name of variable to extract. If NULL, then get all
id	character. Uniquely named dataset identifier
varmeta	should variable metadata be appended?

Value

data.frame

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

read_ftp *Read from FTP*

Description

Read from FTP

Usage

```
read_ftp(URL, cat, lyrs = 1, AOI, ext = NULL, crs = NULL, dates = NULL)
```

Arguments

URL	Unique Resource Identifier (http or local)
cat	catalog element
lyrs	lyrs to extract
AOI	Area of Interest
ext	extent of source (if needed)
crs	crs of source (if needed)
dates	dates of data

Value

SpatRaster

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), time_meta(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

---

read_live_catalog *Read Live Catalog from Github release*

Description

Every month, our data catalog is refreshed. This function reads the most current catalog from the Github release.

Usage

```
read_live_catalog(
  url = paste0("https://github.com/mikejohnson51/climateR-catalogs",
    "/releases/latest/download/", "catalog.parquet")
)
```

Arguments

url	URL to read

Value

data.frame

See Also

Other catalog: catalog

read_zarr_file *Read Zarr File*

Description

Reads a Zarr file from a specified URL and extracts metadata and variable information.

Usage

```
read_zarr_file(URL, varname = NULL, id, varmeta = TRUE)
```

Arguments

URL	Character. The URL of the Zarr file.
varname	Character. Variable name to extract. Defaults to NULL.
id	Character. An identifier for the dataset.
varmeta	Logical. Whether to include variable metadata. Defaults to TRUE.

Value

A data frame with merged metadata and variable information.

See Also

Other zarr: `.resource_grid_zarr()`, `.resource_time_zarr()`, `go_get_zarr()`, `zarr_crop()`, `zarr_get()`, `zarr_to_terra()`, `zarr_xyzv()`

time_meta *Find DAP time metadata*

Description

Find DAP time metadata

Usage

```
time_meta(raw)
```

Arguments

raw	data.frame

Value

data.frame

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), try_att(), var_to_terra(), variable_meta(), vrt_crop_get()

---

try_att *TryCatch around RNetCDF::att.get.nc()*

Description

TryCatch around RNetCDF::att.get.nc()

Usage

```
try_att(nc, variable, attribute)
```

Arguments

nc	"NetCDF" object which points to the NetCDF dataset. Found with RNetCDF::open.nc.
variable	ID or name of the variable from which the attribute will be read, or "NC_GLOBAL" for a global attribute.
attribute	Attribute name or ID.

Value

Vector with a data type that depends on the NetCDF variable. For NetCDF variables of type NC_CHAR, the R type is either character or raw, as specified by argument rawchar. For NC_STRING, the R type is character. Numeric variables are read as double precision by default, but the smallest R type that exactly represents each external type is used if fitnum is TRUE.

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), var_to_terra(), variable_meta(), vrt_crop_get()

variable_meta *Find DAP variable metadata*

Description

Find DAP variable metadata

Usage

```
variable_meta(raw, verbose = TRUE)
```

Arguments

raw	data.frame
verbose	emit messages

Value

data.frame

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), vrt_crop_get()

var_to_terra *Variable Array to SpatRast*

Description

Variable Array to SpatRast

Usage

```
var_to_terra(var, dap)
```

Arguments

var	numeric array
dap	dap description

Value

SpatRast

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), variable_meta(), vrt_crop_get()

`vrt_crop_get` *VRT Crop*

Description

VRT Crop

Usage

```
vrt_crop_get(
  URL = NULL,
  catalog = NULL,
  AOI = NULL,
  grid = NULL,
  varname = NULL,
  start = NULL,
  end = NULL,
  toptobottom = FALSE,
  verbose = TRUE
)
```

Arguments

URL	local file path or URL
catalog	subset of open.dap catalog
AOI	an sf of SpatVect point or polygon to extract data for
grid	a list containing an extent (), and crs
varname	variable name to extract (e.g. tmin)
start	for non "dated" items, start can be called by index
end	for non "dated" items, end can be called by index
toptobottom	should data be inverse?
verbose	Should dap_summary be printed?

Value

SpatRaster

See Also

Other dap: .resource_grid(), .resource_time(), climater_dap(), climater_filter(), dap(), dap_crop(), dap_get(), dap_meta(), dap_summary(), dap_to_local(), dap_xyzv(), extract_sites(), get_data(), go_get_dap_data(), grid_meta(), make_ext(), make_vect(), merge_across_time(), parse_date(), read_dap_file(), read_ftp(), time_meta(), try_att(), var_to_terra(), variable_meta()

`writeDodsrc` *Write dodsrc file*

Description

Write a dodsrc file that is valid for a netrc file

Usage

```
writeDodsrc(netrcFile = getNetrcPath(), dodsrcFile = ".dodsrc")
```

Arguments

`netrcFile`	A character. A path to where the netrc file should be.
`dodsrcFile`	The path to the dodsrc file you want to write By default will go to your home directory, which is advised

Value

A character vector containing the netrc file path

See Also

Other netrc: checkDodsrc(), checkNetrc(), getDodsrcPath(), getNetrcPath(), writeNetrc()

`writeNetrc` *Write netrc file*

Description

Write a netrc file that is valid for accessing urs.earthdata.nasa.gov

Usage

```
writeNetrc(
  login,
  password,
  machine = "urs.earthdata.nasa.gov",
  netrcFile = getNetrcPath(),
  overwrite = FALSE
)
```

Arguments

login	A character. Email address used for logging in on earthdata
password	A character. Password associated with the login.
machine	the machine you are logging into
netrcFile	A character. A path to where the netrc file should be written. By default will go to your home directory, which is advised
overwrite	A logical. overwrite the existing netrc file?

Details

The database is accessed with the user's credentials. A netrc file storing login and password information is required. See here. Once set up you must do the following (1) Login to EarthData (2) Go to Applications > Authorized Apps (3) If NASA GESDISC DATA ARCHIVE is not in the Approved Applications list, select APPROVE MORE APPLICATIONS (4) Find NASA GESDISC DATA ARCHIVE and click AUTHORIZE for instruction on how to register and set DataSpace credential.

Value

A character vector containing the netrc file path

See Also

Other netrc: checkDodsrc(), checkNetrc(), getDodsrcPath(), getNetrcPath(), writeDodsrc()

Examples

```
## Not run:
writeNetrc(
  login = "XXX@email.com",
  password = "yourSecretPassword"
)

## End(Not run)
```

---

zarr_crop *Crop Zarr Data to a Spatial and Temporal Subset*

Description

Crops data in a Zarr file based on spatial (AOI) and temporal (start/end) filters.

Usage

```
zarr_crop(
  URL = NULL,
  catalog = NULL,
  AOI = NULL,
  startDate = NULL,
  endDate = NULL,
  start = NULL,
  end = NULL,
  varname = NULL,
  verbose = TRUE
)
```

Arguments

URL	Character. The URL of the Zarr file. Defaults to NULL.
catalog	Data frame. Metadata catalog for the Zarr file. Defaults to NULL.
AOI	Spatial object. Area of interest for cropping. Defaults to NULL.
startDate	Character. Start date for cropping. Defaults to NULL.
endDate	Character. End date for cropping. Defaults to NULL.
start	Numeric. Start index for cropping. Defaults to NULL.
end	Numeric. End index for cropping. Defaults to NULL.
varname	Character. Variable name to crop. Defaults to NULL.
verbose	Logical. Whether to print verbose output. Defaults to TRUE.

Value

A cropped dataset matching the specified criteria.

See Also

Other zarr: `.resource_grid_zarr()`, `.resource_time_zarr()`, `go_get_zarr()`, `read_zarr_file()`, `zarr_get()`, `zarr_to_terra()`, `zarr_xyzv()`

---

zarr_get *Process and retrieve Zarr data as terra objects*

Description

This function processes Zarr resources and converts them into terra spatial objects.

Usage

```
zarr_get(zarr, varname = NULL)
```

Arguments

zarr	A data frame containing details of the Zarr resources to process. Each row should correspond to a single Zarr resource.
varname	Character vector specifying the variable names to retrieve. If 'NULL', retrieves all available variables.

Value

Returns processed data as terra SpatRaster objects or merged data frames, depending on the input and Zarr properties.

See Also

Other zarr: .resource_grid_zarr(), .resource_time_zarr(), go_get_zarr(), read_zarr_file(), zarr_crop(), zarr_to_terra(), zarr_xyzv()

Examples

```
## Not run:
# Example usage (assuming `zarr` is a properly formatted data frame):
# result <- zarr_get(zarr, varname = "temperature")

## End(Not run)
```

---

zarr_to_terra *Convert data to terra SpatRaster or data frame format*

Description

Converts extracted Zarr data into terra SpatRaster objects or data frames based on spatial and temporal dimensions.

Usage

```
zarr_to_terra(var, zarr)
```

Arguments

var	A variable containing extracted Zarr data.
zarr	A data frame containing details of the Zarr resource, including metadata.

Value

Returns a terra SpatRaster object or data frame containing the processed data.

See Also

Other zarr: .resource_grid_zarr(), .resource_time_zarr(), go_get_zarr(), read_zarr_file(), zarr_crop(), zarr_get(), zarr_xyzv()

Examples

```
## Not run:
# Example usage (assuming `var` and `zarr` are properly formatted):
# result <- zarr_to_terra(var, zarr)

## End(Not run)
```

---

zarr_xyzv *Extract Variable Information from Zarr Object*

Description

Extracts variable and coordinate information from a Zarr object.

Usage

```
zarr_xyzv(obj, varname = NULL, varmeta = FALSE)
```

Arguments

obj	Zarr object or file path.
varname	Character. Specific variable name to extract. Defaults to NULL.
varmeta	Logical. Whether to include variable metadata. Defaults to FALSE.

Value

A data frame containing variable metadata and coordinate information.

See Also

Other zarr: .resource_grid_zarr(), .resource_time_zarr(), go_get_zarr(), read_zarr_file(), zarr_crop(), zarr_get(), zarr_to_terra()

Index