

Package: PEcAn.data.atmosphere (via r-universe)

June 27, 2024

Type Package

Title PEcAn Functions Used for Managing Climate Driver Data

Version 1.7.2

Date 2021-10-04

Description The Predictive Ecosystem Carbon Analyzer (PEcAn) is a scientific workflow management tool that is designed to simplify the management of model parameterization, execution, and analysis. The PEcAn.data.atmosphere package converts climate driver data into a standard format for models integrated into PEcAn. As a standalone package, it provides an interface to access diverse climate data sets.

Imports abind (>= 1.4.5), amerifluxr, assertthat, arrow, curl, dplyr (>= 0.8.1), geonames (> 0.998), ggplot2, glue, httr, jsonlite, lubridate (>= 1.6.0), MASS, mgcv, ncdf4 (>= 1.15), nneo, PEcAn.DB, PEcAn.logger, PEcAn.remote, PEcAn.utils, purrr (>= 0.2.3), raster, REddyProc, reshape2, rlang (>= 0.2.0), sf, sp, stringr (>= 1.1.0), suntools, terra, tibble, tidyr, tidyselect, truncnorm, units, XML (>= 3.98-1.4), xts, zoo

Suggests doParallel, foreach, furr, future, mockery, parallel, PEcAn.settings, progress, reticulate, testthat (>= 2.0.0), withr

Remotes github::adokter/suntools, github::chuhousen/amerifluxr, github::ropensci/geonames, github::ropensci/nneo

License BSD_3_clause + file LICENSE

Copyright Authors

LazyLoad yes

LazyData FALSE

Encoding UTF-8

RoxygenNote 7.3.1

Repository <https://pecanproject.r-universe.dev>

RemoteUrl <https://github.com/PecanProject/pecan>

RemoteRef HEAD

RemoteSha d5c7bffdf233077968945a182c11240b5d76e42d

Contents

AirDens	4
align.met	5
browndog.met	6
build_cf_variables_table_url	7
cfmet.downscale.daily	8
cfmet.downscale.subdaily	9
cfmet.downscale.time	9
check_met_input_file	10
check_unit	11
closest_xy	11
col2ncvar	12
cos_solar_zenith_angle	12
daygroup	13
debias.met.regression	13
debias_met	16
download.Ameriflux	17
download.AmerifluxLBL	17
download.CRUNCEP	19
download.ERA5.old	20
download.FACE	21
download.Fluxnet2015	22
download.FluxnetLaThuile	23
download.Geostreams	24
download.GFDL	25
download.GLDAS	26
download.ICOS	27
download.MACA	28
download.MERRA	29
download.MsTMIP_NARR	30
download.NARR	31
download.NARR_site	32
download.NEONmet	33
download.NLDAS	34
download.NOAA_GEFS	35
download.PalEON	37
download.PalEON_ENS	37
download.raw.met.module	38
download.US_WCr	40
download.US_Wlef	40
download_NOAA_GEFS_EFI	41
downscale_one_cfmet_day	42

downscale_repeat_6hr_to_half_hrly	42
downscale_repeat_6hr_to_hrly	43
downscale_ShortWave_to_half_hrly	43
downscale_ShortWave_to_hrly	44
downscale_solar_geom	45
downscale_solar_geom_halfhour	45
downscale_spline_to_half_hrly	46
downscale_spline_to_hrly	46
equation_of_time	47
ERA5_met_process	47
exner	48
extract.local.CMIP5	49
extract.local.NLDAS	50
extract.nc	51
extract.nc.ERA5	52
gen.subdaily.models	53
generate_narr_url	55
get.es	55
get.lv	56
get.ncvector	56
get.rh	57
get.vpd	58
get_cf_variables_table	58
get_clowderauth	59
get_NARR_thredds	59
get_narr_url	60
half_hour_downscale	61
latlon2lcc	61
latlon2narr	62
lightME	63
lm_ensemble_sims	63
load.cfmet	65
merge_met_variable	66
met.process	67
met.process.stage	68
met2CF.ALMA	69
met2CF.Ameriflux	69
met2CF.AmerifluxLBL	70
met2CF.csv	71
met2CF.ERA5	73
met2CF.FACE	74
met2CF.Geostreams	75
met2CF.ICOS	76
met2CF.NARR	77
met2CF.PalEON	78
met2CF.PalEONregional	78
metgapfill	79
metgapfill.NOAA_GEFS	80

met_temporal_downscale.Gaussian_ensemble	81
model.train	82
narr_flux_vars	83
nc.merge	83
noaa_grid_download	85
noaa_stage2	85
par2ppfd	86
pecan_standard_met_table	87
permute.nc	87
post_process	88
predict_subdaily_met	88
prepare_narr_year	90
process_gridded_noaa_download	91
qair2rh	92
read.register	93
read_narr_var	93
rh2qair	94
save.betas	94
save.model	95
site.lst	96
site_from_tag	96
solarMJ2ppfd	97
spin.met	98
split_wind	99
step_means	100
subdaily_pred	101
sw2par	102
sw2ppfd	102
temporal_downscale.functions	103
temporal_downscale	104
upscale_met	105
wide2long	105
write_noaa_gefs_netcdf	106

Index**108**

AirDens

*Air Density***Description**

estimate air density from pressure, temperature, and humidity

Usage

AirDens(pres, T, rv)

Arguments

pres	air pressure (pascals)
T	air temperature (Kelvin)
rv	humidity

Author(s)

Mike Dietze

align.met

align.met

Description

This script aligns meteorology datasets in at temporal resolution for debiasing & temporal down-scaling. Note: The output here is stored in memory! Note: can probably at borrow from or adapt align_data.R in Benchmarking module, but it's too much of a black box at the moment.

Usage

```
align.met(
  train.path,
  source.path,
  yrs.train = NULL,
  yrs.source = NULL,
  n.ens = NULL,
  pair.mems = FALSE,
  mems.train = NULL,
  seed = Sys.Date(),
  print.progress = FALSE
)
```

Arguments

train.path	- path to the dataset to be used to downscale the data
source.path	- data to be bias-corrected aligned with training data (from align.met)
yrs.train	- (optional) specify a specific years to be loaded for the training data; prevents needing to load the entire dataset. If NULL, all available years will be loaded. If not null, should be a vector of numbers (so you can skip problematic years)
yrs.source	- (optional) specify a specific years to be loaded for the source data; prevents needing to load the entire dataset. If NULL, all available years will be loaded. If not null, should be a vector of numbers (so you can skip problematic years)
n.ens	- number of ensemble members to generate and save
pair.mems	- logical stating whether ensemble members should be paired in the case where ensembles are being read in in both the training and source data

`mems.train` - (optional) string of ensemble identifiers that ensure the training data is read in a specific order to ensure consistent time series & proper error propagation. If null, members of the training data ensemble will be randomly selected and ordered. Specifying the ensemble members IDs (e.g. `CCSM_001`, `CCSM_002`) will ensure ensemble members are properly identified and combined.

`seed` - specify seed so that random draws can be reproduced

`print.progress` - if TRUE, prints progress bar

Details

Align meteorology datasets for debiasing

1. Assumes that both the training and source data are in **at least** daily resolution and each dataset is in a consistent temporal resolution being read from a single file (CF/Pecan format). For example, CMIP5 historical/p1000 runs where radiation drivers are in monthly resolution and temperature is in daily will need to be reconciled using one of the "met2CF" or "download" or "extract" functions

2. Default file structure: Ensembles members for a given site or set of sites are housed in a common folder with the site ID. Right now everything is based off of Christy's PaleON ensemble ID scheme where the site ID is a character string (e.g. HARVARD) followed the SOURCE data family (i.e. GCM) as a string and then the ensemble member ID as a number (e.g. 001). For example, the file path for a single daily ensemble member for PaleON is: `"~/Desktop/Research/met_ensembles/data/met_ensembles/HARVARD/day/ensembles/bcc-csm1-1_004"` with each year in a separate netcdf file inside of it. "bcc-csm1-1_004" is an example of an ensemble member ID that might be used if you are specifying `mems.train`.

Value

2-layered list (stored in memory) containing the training and source data that are now matched in temporal resolution have the specified number of ensemble members - `dat.train` (training dataset) and `dat.source` (source data to be downscaled or bias-corrected) are both lists that contain separate data frames for time indices and all available met variables with ensemble members in columns

Author(s)

Christy Rollinson

See Also

Other debias - Debias & Align Meteorology Datasets into continuous time series: [debias.met.regression\(\)](#)

browndog.met

Use browndog to get the met data for a specific model

Description

Use browndog to get the met data for a specific model

Usage

```
browndog.met(  
  browndog,  
  source,  
  site,  
  start_date,  
  end_date,  
  model,  
  dir,  
  username,  
  con  
)
```

Arguments

browndog	list with url, username and password to connect to browndog
source	the source of the met data, currently only NARR an Ameriflux is supported
site	site information should have id, lat, lon and name (ameriflux id)
start_date	start date for result
end_date	end date for result
model	model to convert the met data to
dir	folder where results are stored (in subfolder)
username	used when downloading data from Ameriflux like sites
con	database connection

Author(s)

Rob Kooper

build_cf_variables_table_url

Construct a URL to a specific version of the CF variables table XML

Description

This uses [sprintf](#) to construct the URL with the version number as the first argument.

Usage

```
build_cf_variables_table_url(  
  version,  
  url_format_string = paste0("http://cfconventions.org/",  
    "Data/cf-standard-names/%d/src/", "src-cf-standard-name-table.xml")  
)
```

Arguments

version CF variables table version number (integer/numeric)
url_format_string A format string passed to `sprintf`. This should contain the entire target URL with the version number replaced by "%d", and *no other string substitutions*.

Value

Complete URL, as a string

Author(s)

Alexey Shiklomanov

cfmet.downscale.daily *Simple, Fast Daily to Hourly Climate Downscaling*

Description

Based on weach family of functions but 5x faster than weachNEW, and requiring metric units (temperature in Kelvins on input and celsius on output, windspeed in kph, precip in mm, relative humidity as fraction). Derived from the weachDT function in the BioCro package.

Usage

```
cfmet.downscale.daily(dailymet, output.dt = 1, lat)
```

Arguments

dailymet data frame with climate variables
output.dt output timestep
lat latitude (for calculating solar radiation)

Value

weather file with subdaily timesteps

Author(s)

David LeBauer

`cfmet.downscale.subdaily`*Subdaily to hourly (or less) downscaling*

Description

Uses simple spline to interpolate variables with diurnal variability, otherwise uses averaging or repeating for variables with no clear diurnal pattern. For all variables except temperature, negative values are set to zero.

Usage

```
cfmet.downscale.subdaily(subdaily, output.dt = 1)
```

Arguments

<code>subdaily</code>	data frame with climate variables queried from load.cfmet
<code>output.dt</code>	output timestep. default is one hour

Value

weather file with subdaily met variables rescaled to output time step

Author(s)

David LeBauer

`cfmet.downscale.time` *Temporal downscaling of daily or subdaily CF met data*

Description

Temporal downscaling of daily or subdaily CF met data

Usage

```
cfmet.downscale.time(cfmet, output.dt = 1, lat = lat, ...)
```

Arguments

<code>cfmet</code>	data frame with CF variables generated by load.cfmet
<code>output.dt</code>	time step (hours) for output
<code>lat</code>	latitude (for calculating solar radiation)
<code>...</code>	ignored

Value

downscaled result

Author(s)

David LeBauer

check_met_input_file *Check a meteorology data file for compliance with the PEcAn standard*

Description

Check a meteorology data file for compliance with the PEcAn standard

Usage

```
check_met_input_file(  
  metfile,  
  variable_table = pecan_standard_met_table,  
  required_vars = variable_table %>% dplyr::filter(.data$is_required) %>%  
    dplyr::pull("cf_standard_name"),  
  warn_unknown = TRUE  
)
```

Arguments

metfile	Path of met file to check, as a scalar character.
variable_table	'data.frame' linking standard names to their units. Must contain columns "cf_standard_name" and "units". Default is [pecan_standard_met_table].
required_vars	Character vector of required variables. Defaults to variables marked as required in 'variable_table'.
warn_unknown	Logical. If 'TRUE' (default), throw a warning for variables not in 'variable_table'. Otherwise, ignore unknown variables.

Value

'data.frame' summarizing the results of the tests.

Author(s)

Alexey Shiklomanov

check_unit	<i>Check that the unit of a variable in a NetCDF file is equivalent to the expected unit.</i>
------------	---

Description

Check that the unit of a variable in a NetCDF file is equivalent to the expected unit.

Usage

```
check_unit(variable, nc, variable_table, warn_unknown = TRUE)
```

Arguments

variable	Name of target variable, as a length 1 character
nc	NetCDF object containing target variable
variable_table	'data.frame' linking standard names to their units. Must contain columns "cf_standard_name" and "units". Default is [pecan_standard_met_table].
warn_unknown	Logical. If 'TRUE' (default), throw a warning for variables not in 'variable_table'. Otherwise, ignore unknown variables.

Value

'TRUE' if unit is correct, or 'try-error' object if there is a mismatch.

Author(s)

Alexey Shiklomanov

closest_xy	<i>closest_xy</i>
------------	-------------------

Description

Given latitude and longitude coordinates, find NARR x and y indices

Usage

```
closest_xy(slat, slon, infolder, infile)
```

Author(s)

Betsy Cowdery, Ankur Desai

col2ncvar	<i>Create 'ncvar' object from variable name</i>
-----------	---

Description

Create 'ncvar' object from variable name

Usage

```
col2ncvar(variable, dims)
```

Arguments

variable	CF variable name
dims	List of NetCDF dimension objects (passed to 'ncdf4::ncvar_def(..., dim)')

Value

'ncvar' object (from 'ncvar_def')

cos_solar_zenith_angle	<i>Cosine of Solar Zenith Angle</i>
------------------------	-------------------------------------

Description

Calculates the cosine of the solar zenith angle based on the given parameters. This angle is crucial in determining the amount of solar radiation reaching a point on Earth.

Usage

```
cos_solar_zenith_angle(doy, lat, lon, dt, hr)
```

Arguments

doy	Day of year. Integer representing the day of the year (1-365).
lat	Latitude in degrees. Positive for the Northern Hemisphere and negative for the Southern Hemisphere.
lon	Longitude in degrees. Positive for East and negative for West.
dt	Time interval in seconds. Represents the duration over which the measurement is averaged or integrated.
hr	Hour of the day (0-23). Specifies the specific hour for which the calculation is made.

Details

For explanations of formulae, see <https://web.archive.org/web/20180307133425/http://www.itacanet.org/the-sun-as-a-source-of-energy/part-3-calculating-solar-angles/>

Value

Numeric value representing the cosine of the solar zenith angle.

Author(s)

Alexey Shiklomanov

References

"Understanding Solar Position and Solar Radiation" - RAMMB: [Link](<https://rammb.cira.colostate.edu/wmovl/vrl/tutorials/>)

Examples

```
cos_solar_zenith_angle(doy = 150, lat = 45, lon = -93, dt = 3600, hr = 12)
```

daygroup

Assign daygroup tag for a given date

Description

Assign daygroup tag for a given date

Usage

```
daygroup(date, flx)
```

debias.met.regression *debias.met.regression*

Description

This script debiases one dataset (e.g. GCM, re-analysis product) given another higher resolution product or empirical observations. It assumes input are in annual CF standard files that are generate from the pecan extract or download funcitons.

Usage

```

debias.met.regression(
  train.data,
  source.data,
  n.ens,
  vars.debias = NULL,
  CRUNCEP = FALSE,
  pair.anoms = TRUE,
  pair.ens = FALSE,
  uncert.prop = "mean",
  resids = FALSE,
  seed = Sys.Date(),
  outfolder,
  yrs.save = NULL,
  ens.name,
  ens.mems = NULL,
  force.sanity = TRUE,
  sanity.tries = 25,
  sanity.sd = 8,
  lat.in,
  lon.in,
  save.diagnostics = TRUE,
  path.diagnostics = NULL,
  parallel = FALSE,
  n.cores = NULL,
  overwrite = TRUE,
  verbose = FALSE
)

```

Arguments

<code>train.data</code>	- training data coming out of <code>align.met</code>
<code>source.data</code>	- data to be bias-corrected aligned with training data (from <code>align.met</code>)
<code>n.ens</code>	- number of ensemble members to generate and save for EACH source ensemble member
<code>vars.debias</code>	- which met variables should be debiased? if <code>NULL</code> , all variables in <code>train.data</code>
<code>CRUNCEP</code>	- flag for if the dataset being downscaled is <code>CRUNCEP</code> ; if <code>TRUE</code> , special cases triggered for met variables that have been naively gapfilled for certain time periods
<code>pair.anoms</code>	- logical stating whether anomalies from the same year should be matched or not
<code>pair.ens</code>	- logical stating whether ensembles from train and source data need to be paired together (for uncertainty propagation)
<code>uncert.prop</code>	- method for error propagation for child ensemble members 1 ensemble member; options=c(random, mean); randomly strongly encouraged if <code>n.ens>1</code>
<code>resids</code>	- logical stating whether to pass on residual data or not *Not implemented yet

seed	- specify seed so that random draws can be reproduced
outfolder	- directory where the data should go
yrs.save	- what years from the source data should be saved; if NULL all years of the source data will be saved
ens.name	- what is the name that should be attached to the debiased ensemble
ens.mems	- what labels/numbers to attach to the ensemble members so we can gradually build bigger ensembles without having to do do giant runs at once; if NULL will be numbered 1:n.ens
force.sanity	- (logical) do we force the data to meet sanity checks?
sanity.tries	- how many time should we try to predict a reasonable value before giving up? We don't want to end up in an infinite loop
sanity.sd	- how many standard deviations from the mean should be used to determine sane outliers (default 8)
lat.in	- latitude of site
lon.in	- longitude of site
save.diagnostics	- logical; save diagnostic plots of output?
path.diagnostics	- path to where the diagnostic graphs should be saved
parallel	- (experimental) logical stating whether to run temporal_downscale_functions.R in parallel *Not Implemented yet
n.cores	- (experimental) how many cores to use in parallelization *Not implemented yet
overwrite	- overwrite existing files? Currently ignored
verbose	logical: should <code>ncdf4</code> functions print debugging information as they run?

Details

Debias Meteorology using Multiple Linear Regression Statistically debias met datasets and generate ensembles based on the observed uncertainty

Author(s)

Christy Rollinson

See Also

Other debias - Debias & Align Meteorology Datasets into continuous time series: [align.met\(\)](#)

debias_met

debias_met

Description

debias.met takes input_met and debiases it based on statistics from a train_met dataset

Usage

```
debias.met(  
  outfolder,  
  input_met,  
  train_met,  
  site_id,  
  de_method = "linear",  
  overwrite = FALSE,  
  verbose = FALSE,  
  ...  
)
```

Arguments

outfolder	location where output is stored
input_met	- the source_met dataset that will be altered by the training dataset in NC format.
train_met	- the observed dataset that will be used to train the modeled dataset in NC format
site_id	BETY site id
de_method	- select which debias method you would like to use, options are 'normal', 'linear regression'
overwrite	logical: replace output file if it already exists? Currently ignored.
verbose	logical: should ncdf4
...	other inputs functions print debugging information as they run?

Author(s)

James Simkins

download.Ameriflux *download.Ameriflux*

Description

Download Ameriflux L2 netCDF files

Usage

```
download.Ameriflux(
  sitename,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

sitename	the FLUXNET ID of the site to be downloaded, used as file name prefix. The 'SITE_ID' field in list of Ameriflux sites
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded. Format is YYYY-MM-DD (will only use the year part of the date)
end_date	the end date of the data to be downloaded. Format is YYYY-MM-DD (will only use the year part of the date)
overwrite	should existing files be overwritten
verbose	should the function be very verbose

Author(s)

Josh Mantooth, Rob Kooper, Ankur Desai

download.AmerifluxLBL *Download Ameriflux LBL CSV files*

Description

download.AmerifluxLBL. Function uses amf_download_base function from amerifluxr package to download a zip-file of data. The zip-file is extracted to a csv-file that is stored to the given outfolder. Details about amf_download_base function can be found here: https://github.com/chuhousen/amerifluxr/blob/master/R/amf_

Usage

```
download.AmerifluxLBL(
  sitename,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  username = "pecan",
  method,
  useremail = "@",
  data_product = "BASE-BADM",
  data_policy = "CCBY4.0",
  ...
)
```

Arguments

sitename	the Ameriflux ID of the site to be downloaded, used as file name prefix. The 'SITE_ID' field in list of Ameriflux sites
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded. Format is YYYY-MM-DD (will only use the year part of the date)
end_date	the end date of the data to be downloaded. Format is YYYY-MM-DD (will only use the year part of the date)
overwrite	should existing files be overwritten
verbose	should the function be very verbose
username	Ameriflux username
method	Optional. download_file() function option. Use this to set custom programs such as ncftp
useremail	Used email, should include 'address sign' for code to be functional
data_product	AmeriFlux data product
data_policy	Two possible licenses (based on the site): 'CCBY4.0' or 'LEGACY'

Details

Uses Ameriflux LBL JSON API to download met data from Ameriflux towers in CSV format

Author(s)

Ankur Desai, Henri Kajasilta based on download.Ameriflux.R by Josh Mantooth, Rob Kooper, Shawn Serbin

Examples

```
## Not run:
result <- download.AmerifluxLBL("US-Akn", "~/", "2011-01-01", "2011-12-31", overwrite=TRUE)

## End(Not run)
```

download.CRUNCEP	<i>Download CRUNCEP data</i>
------------------	------------------------------

Description

Download and convert to CF CRUNCEP single grid point from MSTIMIP server using OPENDAP interface

Usage

```
download.CRUNCEP(
  outfolder,
  start_date,
  end_date,
  lat.in,
  lon.in,
  overwrite = FALSE,
  verbose = FALSE,
  maxErrors = 10,
  sleep = 2,
  method = "ncss",
  ...
)
```

Arguments

outfolder	Directory where results should be written
start_date, end_date	Range of years to retrieve. Format is YYYY-MM-DD, but only the year portion is used and the resulting files always contain a full year of data.
lat.in	site latitude in decimal degrees
lon.in	site longitude in decimal degrees
overwrite	logical. Download a fresh version even if a local file with the same name already exists?
verbose	logical. Passed on to ncvar_def and nc_create to control printing of debug info
maxErrors	Maximum times to re-try following an error accessing netCDF data through THREDDS

sleep	Wait time between attempts following a THREDDS or other error
method	(string) Data access method. 'opendap' (default) attempts to directly access files via OpenDAP. 'ncss' (NetCDF subset) subsets the file on the server, downloads the subsetted file to 'tempfile' and then reads it locally. 'opendap' is faster when it works, but often fails because of server issues. 'ncss' can be much slower, but is more reliable.
...	Other arguments, currently ignored

Author(s)

James Simkins, Mike Dietze, Alexey Shiklomanov

download.ERA5.old *Download ERA 5 data*

Description

Link to [full data documentation](<https://confluence.ecmwf.int/display/CKB/ERA5+data+documentation>).

Usage

```
download.ERA5.old(
    outfolder,
    start_date,
    end_date,
    lat.in,
    lon.in,
    product_types = "all",
    overwrite = FALSE,
    reticulate_python = NULL,
    ...
)
```

Arguments

outfolder	Directory where results should be written
start_date, end_date	Range of years to retrieve. Format is 'YYYY-MM-DD'.
lat.in, lon.in	Site coordinates, decimal degrees (numeric)
product_types	Character vector of product types, or "all". Must be one or more of: "reanalysis", "ensemble members", "ensemble mean", "ensemble spread"
overwrite	Logical. If 'FALSE' (default), skip any files with the same target name (i.e. same variable) that already exist in 'outfolder'. If 'TRUE', silently overwrite existing files.
reticulate_python	Path to Python binary for 'reticulate' (passed to [reticulate::use_python()]). If 'NULL' (default), use the system default.
...	Currently unused. Allows soaking up additional arguments to other methods.

Details

Under the hood, this function uses the Python ‘cdsapi’ module, which can be installed via ‘pip’ (‘pip install –user cdsapi’). The module is accessed via the ‘reticulate’ package.

Using the CDS API requires you to create a free account at <https://cds.climate.copernicus.eu>. Once you have done that, you will need to configure the CDS API on your local machine by creating a ‘\$HOME/.cdsapi’ file, as described [here](<https://cds.climate.copernicus.eu/api-how-to#install-the-cds-api-key>).

Value

Character vector of file names containing raw, downloaded data (invisibly)

Author(s)

Alexey Shiklomanov

Examples

```
## Not run:
files <- download.ERA5(
  "ERA5_output",
  start_date = "2010-01-01",
  end_date = "2010-02-01",
  lat.in = 45.5594,
  lon.in = -84.6738,
  product_types = "all"
)

## End(Not run)
```

download.FACE

Download Raw FACE data from the internet

Description

Download Raw FACE data from the internet

Usage

```
download.FACE(
  sitename,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  method,
  ...
)
```

Arguments

sitename	sitename
outfolder	location where output is stored
start_date	desired start date YYYY-MM-DD
end_date	desired end date YYYY-MM-DD
overwrite	overwrite existing files? Default is FALSE
method	Optional. Passed to download_file() function. Use this to set custom programs such as ncfp to use when downloading files from FTP sites
...	other inputs

Author(s)

Betsy Cowdery

download.Fluxnet2015 *download.Fluxnet2015*

Description

Download Fluxnet 2015 CSV files

Usage

```
download.Fluxnet2015(
  sitename,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  username = "pecan",
  ...
)
```

Arguments

sitename	the FLUXNET ID of the site to be downloaded, used as file name prefix. The 'SITE_ID' field in list of Ameriflux sites
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded. Format is YYYY-MM-DD (will only use the year part of the date)
end_date	the end date of the data to be downloaded. Format is YYYY-MM-DD (will only use the year part of the date)
overwrite	should existing files be overwritten
verbose	should the function be very verbose

Author(s)

Ankur Desai, based on download.Ameriflux.R by Josh Mantooth, Rob Kooper

download.FluxnetLaThuile

download.FluxnetLaThuile

Description

Download Fluxnet LaThuile CSV files

Usage

```
download.FluxnetLaThuile(  
  sitename,  
  outfolder,  
  start_date,  
  end_date,  
  overwrite = FALSE,  
  verbose = FALSE,  
  username = "pecan",  
  ...  
)
```

Arguments

sitename	the FLUXNET ID of the site to be downloaded, used as file name prefix. The 'SITE_ID' field in list of Fluxnet LaThuile sites
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded. Format is YYYY-MM-DD (will only use the year part of the date)
end_date	the end date of the data to be downloaded. Format is YYYY-MM-DD (will only use the year part of the date)
overwrite	should existing files be overwritten
verbose	should the function be very verbose
username	should be the registered Fluxnet username, else defaults to pecan

Author(s)

Ankur Desai

download.Geostreams *Download Geostreams data from Clowder API*

Description

Download Geostreams data from Clowder API

Usage

```
download.Geostreams(
    outfolder,
    sitename,
    start_date,
    end_date,
    url = "https://terraref.ncsa.illinois.edu/clowder/api/geostreams",
    key = NULL,
    user = NULL,
    pass = NULL,
    ...
)
```

Arguments

outfolder	directory in which to save json result. Will be created if necessary
sitename	character. Must match a Geostreams sensor_name
start_date, end_date	datetime
url	base url for Clowder host
key, user, pass	authentication info for Clowder host.
...	other arguments passed as query parameters

Details

Depending on the setup of your Clowder host, authentication may be by username/password, by API key, or skipped entirely. `download.Geostreams` looks first in its call arguments for an API key, then a username and password, then if these are NULL it looks in the user's home directory for a file named `'~/pecan.clowder.xml'`, and finally if no keys or passwords are found there it attempts to connect unauthenticated.

If using `'~/pecan.clowder.xml'`, it must be a valid PEcAn-formatted XML settings file and must contain a `<clowder>` key that specifies hostname, user, and password for your Clowder server:

```
<?xml version="1.0"?>
<pecan>
  <clowder>
    <hostname>terraref.ncsa.illinois.edu</hostname>
```



```

    <user>yourname</user>
    <password>superSecretPassw0rd</password>
  </clowder>
</pecan>

```

Author(s)

Harsh Agrawal, Chris Black

Examples

```

## Not run:
download.Geostreams(outfolder = "~/output/dbfiles/Clowder_EF",
                    sitename = "UIUC Energy Farm - CEN",
                    start_date = "2016-01-01", end_date="2016-12-31",
                    key="verysecret")

## End(Not run)

```

download.GFDL	<i>Download GFDL CMIP5 outputs for a single grid point using OPeNDAP and convert to CF</i>
---------------	--

Description

Download GFDL CMIP5 outputs for a single grid point using OPeNDAP and convert to CF

Usage

```

download.GFDL(
  outfolder,
  start_date,
  end_date,
  lat.in,
  lon.in,
  overwrite = FALSE,
  verbose = FALSE,
  model = "CM3",
  scenario = "rcp45",
  ensemble_member = "r1i1p1",
  ...
)

```

Arguments

outfolder	Directory for storing output
start_date	Start date for met (will be converted via [base::as.POSIXlt])
end_date	End date for met (will be converted via [base::as.POSIXlt])

lat.in	Latitude coordinate for met
lon.in	Longitude coordinate for met
overwrite	Logical: Download a fresh version even if a local file with the same name already exists?
verbose	Logical, passed on to <code>ncvar_def</code> and <code>nc_create</code> to control printing of debug info
model	Which GFDL model to run (options are CM3, ESM2M, ESM2G)
scenario	Which scenario to run (options are rcp26, rcp45, rcp60, rcp85)
ensemble_member	Which ensemble_member to initialize the run (options are r1i1p1, r3i1p1, r5i1p1)

Author(s)

James Simkins, Alexey Shiklomanov, Ankur Desai

download.GLDAS *Download GLDAS data*

Description

Download and convert single grid point GLDAS to CF single grid point from hydro1.sci.gsfc.nasa.gov using OPENDAP interface

Usage

```
download.GLDAS(
  outfolder,
  start_date,
  end_date,
  site_id,
  lat.in,
  lon.in,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

outfolder	location where output is stored
start_date	desired start date
end_date	desired end date
site_id	desired site id
lat.in	latitude of site

lon.in	longitude of site
overwrite	overwrite existing files? Default is FALSE
verbose	Default is FALSE, used as input for ncdf4::ncvar_def
...	other inputs

Author(s)

Christy Rollinson

download.ICOS	<i>Download ICOS Ecosystem data products</i>
---------------	--

Description

Currently available products: Drought-2018 ecosystem eddy covariance flux product <https://www.icos-cp.eu/data-products/YVR0-4898> ICOS Final Fully Quality Controlled Observational Data (Level 2) <https://www.icos-cp.eu/data-products/ecosystem-release>

Usage

```
download.ICOS(
  sitename,
  outfolder,
  start_date,
  end_date,
  product,
  overwrite = FALSE,
  ...
)
```

Arguments

sitename	ICOS id of the site. Example - "BE-Bra"
outfolder	path to the directory where the output file is stored. If specified directory does not exist, it is created.
start_date	start date of the data request in the form YYYY-MM-DD
end_date	end date area of the data request in the form YYYY-MM-DD
product	ICOS product to be downloaded. Currently supported options: "Drought2018", "ETC"
overwrite	should existing files be overwritten. Default False.
...	used when extra arguments are present.

Value

information about the output file

Author(s)

Ayush Prasad

Examples

```
## Not run:
download.ICOS("FI-Sii", "/home/carya/pecan", "2016-01-01", "2018-01-01", product="Drought2018")

## End(Not run)
```

download.MACA

download.MACA

Description

Download MACA CMIP5 outputs for a single grid point using OPeNDAP and convert to CF

Usage

```
download.MACA(
  outfolder,
  start_date,
  end_date,
  site_id,
  lat.in,
  lon.in,
  model = "IPSL-CM5A-LR",
  scenario = "rcp85",
  ensemble_member = "r1i1p1",
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

outfolder	location where output is stored
start_date	, of the format "YEAR-01-01 00:00:00"
end_date	, of the format "YEAR-12-31 23:59:59"
site_id	BETY site id
lat.in	latitude of site
lon.in	longitude of site
model	, select which MACA model to run (options are BNU-ESM, CNRM-CM5, CSIRO-Mk3-6-0, bcc-csm1-1, bcc-csm1-1-m, CanESM2, GFDL-ESM2M, GFDL-ESM2G, HadGEM2-CC365, HadGEM2-ES365, inmcm4, MIROC5, MIROC-ESM, MIROC-ESM-CHEM, MRI-CGCM3, CCSM4, IPSL-CM5A-LR, IPSL-CM5A-MR, IPSL-CM5B-LR, NorESM1-M)

scenario , select which scenario to run (options are rcp45, rcp85)
 ensemble_member , r1i1p1 is the only ensemble member available for this dataset, CCSM4 uses r6i1p1 instead
 overwrite overwrite existing files? Default is FALSE
 verbose Default is FALSE, used as input in ncd4::ncvar_def
 ... other inputs

Author(s)

James Simkins

download.MERRA *Download MERRA data*

Description

Download MERRA data

Usage

```

download.MERRA(
  outfolder,
  start_date,
  end_date,
  lat.in,
  lon.in,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)

```

Arguments

outfolder Directory where results should be written
 start_date, end_date Range of years to retrieve. Format is YYYY-MM-DD, but only the year portion is used and the resulting files always contain a full year of data.
 lat.in site latitude in decimal degrees
 lon.in site longitude in decimal degrees
 overwrite logical. Download a fresh version even if a local file with the same name already exists?
 verbose logical. Passed on to [ncvar_def](#) and [nc_create](#) to control printing of debug info
 ... Not used – silently soak up extra arguments from ‘convert_input’, etc.

Value

'data.frame' of meteorology data metadata

Author(s)

Alexey Shiklomanov

download.MsTMIP_NARR *download.MsTMIP_NARR*

Description

Download and conver to CF NARR single grid point from MSTIMIP server using OPENDAP interface

Usage

```
download.MsTMIP_NARR(
  outfolder,
  start_date,
  end_date,
  site_id,
  lat.in,
  lon.in,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

outfolder	location where output is stored
start_date	YYYY-MM-DD
end_date	YYYY-MM-DD
site_id	BETY site id
lat.in	latitude of site
lon.in	longitude of site
overwrite	overwrite existing files? Default is FALSE
verbose	Default is FALSE, used in ncdf4::ncvar_def
...	Other inputs

Author(s)

James Simkins

download.NARR	<i>Download NARR files</i>
---------------	----------------------------

Description

Download NARR files

Usage

```
download.NARR(
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  method,
  ...
)
```

Arguments

outfolder	location where output is stored
start_date	desired start date YYYY-MM-DD
end_date	desired end date YYYY-MM-DD
overwrite	Overwrite existing files? Default=FALSE
verbose	Turn on verbose output? Default=FALSE
method	Method of file retrieval. Can set this using the options(download.ftp.method=[method]) in your Rprofile.
...	other inputs example options(download.ftp.method="ncftpget")

Author(s)

Betsy Cowdery, Shawn Serbin

Examples

```
## Not run:
download.NARR("~/", '2000/01/01', '2000/01/02', overwrite = TRUE, verbose = TRUE)

## End(Not run)
```

download.NARR_site *Download NARR time series for a single site*

Description

Download NARR time series for a single site

Usage

```
download.NARR_site(
  outfolder,
  start_date,
  end_date,
  lat.in,
  lon.in,
  overwrite = FALSE,
  verbose = FALSE,
  progress = TRUE,
  parallel = TRUE,
  ncores = if (parallel) parallel::detectCores() else NULL,
  ...
)
```

Arguments

outfolder	Target directory for storing output
start_date	Start date for met data
end_date	End date for met data
lat.in	Site latitude coordinate
lon.in	Site longitude coordinate
overwrite	Overwrite existing files? Default=FALSE
verbose	Turn on verbose output? Default=FALSE
parallel	Download in parallel? Default = TRUE
ncores	Number of cores for parallel download. Default is 'parallel::detectCores()'

Author(s)

Alexey Shiklomanov

Examples

```
## Not run:
download.NARR_site(tempdir(), "2001-01-01", "2001-01-12", 43.372, -89.907)

## End(Not run)
```

download.NEONmet	<i>Download NEON Site Met CSV files</i>
------------------	---

Description

download.NEONmet

Usage

```
download.NEONmet(  
  sitename,  
  outfolder,  
  start_date,  
  end_date,  
  overwrite = FALSE,  
  verbose = FALSE,  
  ...  
)
```

Arguments

sitename	the NEON ID of the site to be downloaded, used as file name prefix. The 4-letter SITE code in list of NEON sites
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded. Format is YYYY-MM-DD (will only use the year and month of the date)
end_date	the end date of the data to be downloaded. Format is YYYY-MM-DD (will only use the year and month part of the date)
overwrite	should existing files be overwritten
verbose	makes the function output more text

Details

Uses NEON v0 API to download met data from NEON towers and convert to CF NetCDF

Examples

```
## Not run:  
result <- download.NEONmet('HARV', '~/ ', '2017-01-01', '2017-01-31', overwrite=TRUE)  
  
## End(Not run)
```

download.NLDAS	<i>Download NLDAS met data</i>
----------------	--------------------------------

Description

Download and convert single grid point NLDAS to CF single grid point from hydro1.sci.gsfc.nasa.gov using OPENDAP interface

Usage

```
download.NLDAS(  
  outfolder,  
  start_date,  
  end_date,  
  site_id,  
  lat.in,  
  lon.in,  
  overwrite = FALSE,  
  verbose = FALSE,  
  ...  
)
```

Arguments

outfolder	location of output
start_date	desired start date YYYY-MM-DD
end_date	desired end date YYYY-MM-DD
site_id	site id (BETY)
lat.in	latitude of site
lon.in	longitude of site
overwrite	overwrite existing files? Default is FALSE
verbose	Turn on verbose output? Default=FALSE
...	Other inputs

Author(s)

Christy Rollinson (with help from Ankur Desai)

download.NOAA_GEFS *Download NOAA GEFS Weather Data*

Description

Download NOAA GEFS Weather Data

Usage

```
download.NOAA_GEFS(
  site_id,
  sitename = NULL,
  username = "pecan",
  lat.in,
  lon.in,
  outfolder,
  start_date = Sys.Date(),
  end_date = start_date + lubridate::days(16),
  downscale = TRUE,
  overwrite = FALSE,
  ...
)
```

Arguments

site_id	The unique ID given to each site. This is used as part of the file name.
sitename	Site name
username	username from pecan workflow
lat.in	site latitude in decimal degrees
lon.in	site longitude in decimal degrees
outfolder	Directory where results should be written
start_date	Range of dates/times to be downloaded (default assumed to be time that function is run)
end_date	end date for range of dates to be downloaded (default 16 days from start_date)
downscale	logical, assumed True. Indicated whether data should be downscaled to hourly
overwrite	logical. Download a fresh version even if a local file with the same name already exists?
...	Additional optional parameters

Value

A list of data frames is returned containing information about the data file that can be used to locate it later. Each data frame contains information about one file.

Information on Units

Information on NOAA weather units can be found below. Note that the temperature is measured in degrees C, but is converted at the station and downloaded in Kelvin.

NOAA_GEFS General Information

This function downloads NOAA GEFS weather data. GEFS is an ensemble of 21 different weather forecast models. A 16 day forecast is available every 6 hours. Each forecast includes information on a total of 8 variables. These are transformed from the NOAA standard to the internal PEcAn standard.

Data Availability

NOAA GEFS weather data is available on a rolling 12 day basis; dates provided in "start_date" must be within this range. The end date can be any point after that, but if the end date is beyond 16 days, only 16 days worth of forecast are recorded. Times are rounded down to the previous 6 hour forecast. NOAA GEFS weather data isn't always posted immediately, and to compensate, this function adjusts requests made in the last two hours back two hours (approximately the amount of time it takes to post the data) to make sure the most current forecast is used.

Data Save Format

Data is saved in the netcdf format to the specified directory. File names reflect the precision of the data to the given range of days. NOAA.GEFS.willow creek.3.2018-06-08T06:00.2018-06-24T06:00.nc specifies the forecast, using ensemble number 3 at willow creek on June 6th, 2018 at 6:00 a.m. to June 24th, 2018 at 6:00 a.m.

Author(s)

Quinn Thomas, modified by K Zarada

References

<https://www.ncdc.noaa.gov/crn/measurements.html>

Examples

```
## Not run:
download.NOAA_GEFS(outfolder=~"/Working/results",
  lat.in= 45.805925,
  lon.in = -90.07961,
  site_id = 676)

## End(Not run)
```

download.PaleON	<i>download.PaleON</i>
-----------------	------------------------

Description

Download PaleON files

Usage

```
download.PaleON(
  sitename,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  ...
)
```

Arguments

sitename	sitename
outfolder	desired output location
start_date	desired start date YYYY-MM-DD
end_date	desired end date YYYY-MM-DD
overwrite	overwrite existing files? Default is FALSE
...	Other inputs

Author(s)

Betsy Cowdery

download.PaleON_ENS	<i>Download PaleON met ensemble files</i>
---------------------	---

Description

Download PaleON met ensemble files

Usage

```
download.PalEON_ENS(
  sitename,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  ...
)
```

Arguments

sitename	sitename
outfolder	desired output folder
start_date	desired start date YYYY-MM-DD
end_date	desired end date YYYY-MM-DD
overwrite	overwrite existing files? Default is FALSE
...	Other inputs

Author(s)

Betsy Cowdery, Mike Dietze

download.raw.met.module

download.raw.met.module

Description

download.raw.met.module

Usage

```
.download.raw.met.module(
  dir,
  met,
  register,
  machine,
  start_date,
  end_date,
  str_ns,
  con,
  input_met,
  site.id,
  lat.in,
```

```

    lon.in,
    host,
    site,
    username,
    overwrite = FALSE,
    dbparms,
    Ens.Flag = FALSE
)

```

Arguments

dir	directory to write outputs to
met	source included in input_met
register	register.xml, provided by met.process
machine	machine associated with hostname, provided by met.process
start_date	the start date of the data to be downloaded (will only use the year part of the date)
end_date	the end date of the data to be downloaded (will only use the year part of the date)
str_ns	substitute for site_id if not provided, provided by met.process
con	database connection based on dbparms in met.process
input_met	Which data source to process
site.id	site id
lat.in	site latitude, provided by met.process
lon.in	site longitude, provided by met.process
host	host info from settings file
site	site info from settings file
username	database username
overwrite	whether to force download.raw.met.module to proceed
dbparms	database settings from settings file
Ens.Flag	default set to FALSE

Value

A list of data frames is returned containing information about the data file that can be used to locate it later. Each data frame contains information about one file.

download.US_WCr	<i>download.US-WCr</i>
-----------------	------------------------

Description

download.US-WCr

Usage

```
download.US_WCr(start_date, end_date, timestep = 1)
```

Arguments

start_date	Start date/time data should be downloaded for
end_date	End date/time data should be downloaded for
timestep	How often to take data points from the file. Must be a multiple of 0.5

General Description

Obtains data from Ankur Desai's Willow Creek flux tower, and selects certain variables (NEE and LE) to return Data is returned at the given timestep in the given range.

This data includes information on a number of flux variables.

The timestep parameter is measured in hours, but is then converted to half hours because the data's timestep is every half hour.

Author(s)

Luke Dramko

download.US_Wlief	<i>download.US_Wlief</i>
-------------------	--------------------------

Description

download.US_Wlief

Usage

```
download.US_Wlief(start_date, end_date, timestep = 1)
```

Arguments

start_date	Start date/time data should be downloaded for
end_date	End date/time data should be downloaded for
timestep	How often to take data points from the file. Must be integer

General Description

Obtains data from Ankur Desai's WLEF/ Parks Fall flux tower, and selects certain variables (NEE and LE) to return Data is returned at the given timestep in the given range.

This data includes information on a number of flux variables.

Author(s)

Luke Dramko and K Zarada

download_NOAA_GEFS_EFI

download_NOAA_GEFS_EFI

Description

download_NOAA_GEFS_EFI

Usage

download_NOAA_GEFS_EFI(sitename, outfolder, start_date, site.lat, site.lon)

Arguments

sitename	NEON site name
outfolder	filepath to save ensemble member .nc files
start_date	start date for met forecast
site.lat	site lat
site.lon	site lon

Value

message confirming download complete and location of .nc files

Author(s)

Alexis Helgeson

downscale_one_cfmet_day

Internal helper to downscale a single row from a daily file

Description

Internal helper to downscale a single row from a daily file

Usage

```
downscale_one_cfmet_day(df, tseq, lat)
```

Arguments

df	one row from dailymet
tseq	vector of hours at which to estimate
lat	latitude

Value

df with one row for each hour in 'tseq'

downscale_repeat_6hr_to_half_hrly

Downscale repeat to half hourly

Description

Downscale repeat to half hourly

Usage

```
downscale_repeat_6hr_to_half_hrly(df, varName, hr = 0.5)
```

Arguments

df	dataframe of data to be downscaled (Longwave)
varName	variable names to be downscaled
hr	hour to downscale to- default is 0.5

Value

A dataframe of downscaled data

Author(s)

Laura Puckett

downscale_repeat_6hr_to_hrly
Downscale repeat to hourly

Description

Downscale repeat to hourly

Usage

```
downscale_repeat_6hr_to_hrly(df, varName, hr = 1)
```

Arguments

df	dataframe of data to be downscaled (Longwave)
varName	variable names to be downscaled
hr	hour to downscale to- default is 1

Value

A dataframe of downscaled data

Author(s)

Laura Puckett

downscale_ShortWave_to_half_hrly
Downscale shortwave to half hourly

Description

Downscale shortwave to half hourly

Usage

```
downscale_ShortWave_to_half_hrly(df, lat, lon, hr = 0.5)
```

Arguments

df	data frame of variables
lat	lat of site
lon	long of site
hr	hour to downscale to- default is 1

Value

A dataframe of downscaled state variables

ShortWave.ds

Author(s)

Laura Puckett

downscale_ShortWave_to_hrly

Downscale shortwave to hourly

Description

Downscale shortwave to hourly

Usage

```
downscale_ShortWave_to_hrly(df, lat, lon, hr = 1)
```

Arguments

df	data frame of variables
lat	lat of site
lon	long of site
hr	hour to downscale to- default is 1

Value

A dataframe of downscaled state variables

ShortWave.ds

Author(s)

Laura Puckett

downscale_solar_geom *Calculate potential shortwave radiation*

Description

Calculate potential shortwave radiation

Usage

downscale_solar_geom(doy, lon, lat)

Arguments

doy	day of year in decimal
lon	longitude
lat	latitude

Value

vector of potential shortwave radiation for each doy

Author(s)

Quinn Thomas

downscale_solar_geom_halfhour
Calculate potential shortwave radiation

Description

Calculate potential shortwave radiation

Usage

downscale_solar_geom_halfhour(doy, lon, lat)

Arguments

doy	day of year in decimal
lon	longitude
lat	latitude

Value

vector of potential shortwave radiation for each doy

Author(s)

Quinn Thomas

`downscale_spline_to_half_hrly`*Downscale spline to half hourly*

Description

Downscale spline to half hourly

Usage`downscale_spline_to_half_hrly(df, VarNames, hr = 0.5)`**Arguments**

<code>df</code>	dataframe of data to be downscales
<code>VarNames</code>	variable names to be downscaled
<code>hr</code>	hour to downscale to- default is 0.5

Value

A dataframe of half hourly downscaled state variables

Author(s)

Laura Puckett

`downscale_spline_to_hrly`*Downscale spline to hourly*

Description

Downscale spline to hourly

Usage`downscale_spline_to_hrly(df, VarNames, hr = 1)`**Arguments**

<code>df</code>	dataframe of data to be downscales
<code>VarNames</code>	variable names to be downscaled
<code>hr</code>	hour to downscale to- default is 1

Value

A dataframe of downscaled state variables

Author(s)

Laura Puckett

equation_of_time *Equation of time: Eccentricity and obliquity*

Description

For description of calculations, see https://en.wikipedia.org/wiki/Equation_of_time#Calculating_the_equation_of_time

Usage

```
equation_of_time(doy)
```

Arguments

doy Day of year

Value

'numeric(1)'
length of the solar day, in hours.

Author(s)

Alexey Shiklomanov

ERA5_met_process *Met Processes for ERA5 data*

Description

Met Processes for ERA5 data

Usage

```
ERA5_met_process(settings, in.path, out.path, write.db = FALSE, write = TRUE)
```

Arguments

settings	a multi-settings object
in.path	met input path
out.path	output path
write.db	if write into Bety database
write	if write the settings into pecan.xml file in the outdir of settings.

Value

if write.db is True then return input IDs with physical paths; if write.db is False then return just physical paths of extracted ERA5 clim files.

Author(s)

Dongchen Zhang

exner

Exner function

Description

estimated exner function

Usage

exner(pres)

Arguments

pres	air pressure (Bar)
------	--------------------

Author(s)

Mike Dietze

extract.local.CMIP5 *Extract NLDAS from local download Extract NLDAS meteorology for a point from a local download of the full grid*

Description

This function extracts CMIP5 data from grids that have been downloaded and stored locally. Files are saved as a netCDF file in CF conventions at *DAILY* resolution. Note: At this point in time, variables that are only available at a native monthly resolution will be repeated to give a pseudo-daily record (and can get dealt with in the downscaling workflow). These files are ready to be used in the general PEcAn workflow or fed into the downscaling workflow.

Usage

```
extract.local.CMIP5(
  outfolder,
  in.path,
  start_date,
  end_date,
  lat.in,
  lon.in,
  model,
  scenario,
  ensemble_member = "r1i1p1",
  date.origin = NULL,
  adjust.pr = 1,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

outfolder	- directory where output files will be stored
in.path	- path to the raw full grids
start_date	- first day for which you want to extract met (yyyy-mm-dd)
end_date	- last day for which you want to extract met (yyyy-mm-dd)
lat.in	site latitude in decimal degrees
lon.in	site longitude in decimal degrees
model	which GCM to extract data from
scenario	which experiment to pull (p1000, historical, ...)
ensemble_member	which CMIP5 experiment ensemble member

date.origin	(optional) specify the date of origin for timestamps in the files being read. If NULL defaults to 1850 for historical simulations (except MPI-ESM-P) and 850 for p1000 simulations (plus MPI-ESM-P historical). Format: YYYY-MM-DD
adjust.pr	- adjustment factor fore precipitation when the extracted values seem off
overwrite	logical. Download a fresh version even if a local file with the same name already exists?
verbose	logical. to control printing of debug info
...	Other arguments, currently ignored

Author(s)

Christy Rollinson

extract.local.NLDAS *Extract NLDAS from local download Extract NLDAS meteorology for a point from a local download of the full grid*

Description

This function extracts NLDAS data from grids that have been downloaded and stored locally. Once upon a time, you could query these files directly from the internet, but now they're behind a tricky authentication wall. Files are saved as a netCDF file in CF conventions. These files are ready to be used in the general PEcAn workflow or fed into the downscaling workflow.

Usage

```
extract.local.NLDAS(
  outfolder,
  in.path,
  start_date,
  end_date,
  lat.in,
  lon.in,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

outfolder	- directory where output files will be stored
in.path	- path to the raw full grids
start_date	- first day for which you want to extract met (yyyy-mm-dd)
end_date	- last day for which you want to extract met (yyyy-mm-dd)
lat.in	site latitude in decimal degrees

lon.in	site longitude in decimal degrees
overwrite	logical. Download a fresh version even if a local file with the same name already exists?
verbose	logical. Passed on to <code>ncvar_def</code> and <code>nc_create</code> to control printing of debug info
...	Other arguments, currently ignored

Author(s)

Christy Rollinson

extract.nc	<i>extract.nc</i>
------------	-------------------

Description

Given latitude and longitude coordinates, extract site data from NARR file

Usage

```
extract.nc(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  slat,
  slon,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

in.path	location on disk where inputs are stored
in.prefix	prefix of input files
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be permuted (will only use the year part of the date)
end_date	the end date of the data to be permuted (will only use the year part of the date)
slat	the latitude of the site
slon	the longitude of the site
overwrite	should existing files be overwritten
verbose	should output of function be extra verbose

Author(s)

Betsy Cowdery

extract.nc.ERA5	<i>ERA5_extract</i>
-----------------	---------------------

Description

ERA5_extract

Usage

```
extract.nc.ERA5(
  slat,
  slon,
  in.path,
  start_date,
  end_date,
  outfolder,
  in.prefix,
  newsite,
  vars = NULL,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

slat	latitude
slon	longitude
in.path	path to the directory containing the file to be inserted
start_date	start date
end_date	end date
outfolder	Path to directory where nc files need to be saved.
in.prefix	initial portion of the filename that does not vary by date. Does not include directory; specify that as part of in.path.
newsite	site name.
vars	variables to be extracted. If NULL all the variables will be returned.
overwrite	Logical if files needs to be overwritten.
verbose	Decide if we want to stop printing info.
...	other inputs.

Details

For the list of variables check out the documentation at <https://confluence.ecmwf.int/display/CKB/ERA5+data+documentation+Spatialgrid>

Value

a list of xts objects with all the variables for the requested years

Examples

```
## Not run:
point.data <- ERA5_extract(sslats=40, slons=-120, years=c(1990:1995), vars=NULL)

purrr::map(~xts::apply.daily(.x, mean))

## End(Not run)
```

gen.subdaily.models *gen.subdaily.models*

Description

This is the 2nd function in the tdm workflow that takes the `dat.train_file` that is created from the `nc2dat.train` function and generates "lag.days" and "next.days". These variables pass along information of the previous time step and provides a preview of the next time step. After these variables are created, the models are generated by calling the `tdm_temporal_downscale_functions.R` scripts and these models and betas are saved separately. Please note that these models and betas require a significant amount of space. The storage required varies by the size of the training dataset, but prepare for >100 GB. These will be called later in `tdm_predict_subdaily_met` to perform the linear regression analysis.

Usage

```
gen.subdaily.models(
  outfolder,
  path.train,
  yrs.train,
  direction.filter = "forward",
  in.prefix,
  n.beta,
  day.window,
  seed = Sys.time(),
  resids = FALSE,
  parallel = FALSE,
  n.cores = NULL,
  overwrite = TRUE,
```

```

    verbose = FALSE,
    print.progress = FALSE
  )

```

Arguments

`outfolder` - directory where models will be stored *** storage required varies by size of training dataset, but prepare for >10 GB

`path.train` - path to CF/PEcAn style training data where each year is in a separate file.

`yrs.train` - which years of the training data should be used for to generate the model for the subdaily cycle. If NULL, will default to all years

`direction.filter` - Whether the model will be filtered backward or forward in time. options = c("backward", "forward") (PaleON will go backward, anybody interested in the future will go forward)

`in.prefix`

`n.beta` - number of betas to save from linear regression model

`day.window` - integer specifying number of days around the day being modeled you want to use data from for that specific hours coefficients. Must be integer because we want statistics from the same time of day for each day surrounding the model day

`seed` - seed for randomization to allow for reproducible results

`resids` - logical stating whether to pass on residual data or not (this increases both memory & storage requirements)

`parallel` - logical stating whether to run `temporal_downscale_functions.R` in parallel

`n.cores` - deals with parallelization

`overwrite` logical: replace output file if it already exists?

`verbose` logical, currently ignored

`print.progress` - print progress bar? (gets passed through)

Details

Generate Subdaily Models Create statistical models to predict subdaily meteorology

Author(s)

Christy Rollinson, James Simkins

See Also

Other tdm - Temporally Downscale Meteorology: [lm_ensemble_sims\(\)](#), [model.train\(\)](#), [nc.merge\(\)](#), [predict_subdaily_met\(\)](#), [save.betas\(\)](#), [save.model\(\)](#), [subdaily_pred\(\)](#), [temporal_downscale_functions\(\)](#)

generate_narr_url *Generate NARR url from a vector of dates*

Description

Figures out file names for the given dates, based on NARR's convoluted and inconsistent naming scheme.

Usage

```
generate_narr_url(dates, flx)
```

Arguments

dates	Vector of dates for which to generate URL
flx	(Logical) If 'TRUE', format for 'flx' variables. Otherwise, format for 'sfc' variables. See [narr_flux_vars].

Author(s)

Alexey Shiklomanov

get.es *get es*

Description

Calculate saturation vapor pressure

Usage

```
get.es(temp)
```

Arguments

temp	temperature in degrees C
------	--------------------------

Value

saturation vapor pressure in mb

Author(s)

David LeBauer

Examples

```
temp <- -30:30
plot(temp, get.es(temp))
```

get.lv	<i>Latent heat of vaporization</i>
--------	------------------------------------

Description

calculate latent heat of vaporization for water

Usage

```
get.lv(airtemp = 268.6465)
```

Arguments

airtemp air temperature (Kelvin)

Value

IV latent heat of vaporization (J kg-1)

Author(s)

Istem Fer

get.ncvector	<i>Get time series vector from netCDF file</i>
--------------	--

Description

internal convenience function for streamlining extraction of data from netCDF files with CF-compliant variable names

Usage

```
get.ncvector(var, lati = lati, loni = loni, run.dates = run.dates, met.nc)
```

Arguments

var name of variable to extract
 lati, loni latitude and longitude to extract
 run.dates data frame of dates to read
 met.nc netcdf file with CF variable names

Value

numeric vector

Author(s)

David Shaner LeBauer

get.rh

get RH

Description

Calculate RH from temperature and dewpoint

Usage

`get.rh(T, Td)`

Arguments

T	air temperature, Kelvin
Td	dewpoint, Kelvin

Details

Based on equation 12 in Lawrence 2005, The Relationship between Relative Humidity and the Dew-point Temperature in Moist Air A Simple Conversion and Applications. BAMS <https://doi.org/10.1175/BAMS-86-2-225> $R = 461.5 \text{ K}^{-1} \text{ kg}^{-1}$ gas constant H₂O L enthalpy of vaporization linear dependence on T (p 226, following eq 9)

Value

Relative Humidity numeric vector

Author(s)

David LeBauer

`get.vpd`*VPD*

Description

Calculate VPD

Usage

```
get.vpd(rh, temp)
```

Arguments

rh	relative humidity, in percent
temp	temperature, degrees celsius

Details

Calculate vapor pressure deficit from relative humidity and temperature.

Value

vpd: vapor pressure deficit, in mb

Author(s)

David LeBauer

Examples

```
temp <- -30:30
plot(temp, get.vpd(0, temp))
```

`get_cf_variables_table`

Retrieve the current CF variables table from cfconventions.org and convert it into a data.frame

Description

Retrieve the current CF variables table from cfconventions.org and convert it into a data.frame

Usage

```
get_cf_variables_table(cf_url = build_cf_variables_table_url(57))
```

Arguments

cf_url URL of CF variables table XML. See also [build_cf_variables_table_url](#).

Value

CF variables table, as a tibble

Author(s)

Alexey Shiklomanov

get_clowderauth	<i>Authentication lookup helper</i>
-----------------	-------------------------------------

Description

Authentication lookup helper

Usage

```
get_clowderauth(key, user, pass, url, authfile = "~/pecan.clowder.xml")
```

Arguments

key, user, pass	passed unchanged from download.Geostreams call, possibly null
url	matched against <hostname> in authfile, ignored if authfile contains no host-name.
authfile	path to a PEcAn-formatted XML settings file; must contain a <clowder> key

get_NARR_thredds	<i>Retrieve NARR data using thredds</i>
------------------	---

Description

Retrieve NARR data using thredds

Usage

```
get_NARR_thredds(
  start_date,
  end_date,
  lat.in,
  lon.in,
  progress = TRUE,
  drop_outside = TRUE,
  parallel = TRUE,
  ncores = 1
)
```

Arguments

start_date	Start date for meteorology
end_date	End date for meteorology
lat.in	Latitude coordinate
lon.in	Longitude coordinate
progress	Whether or not to show a progress bar (default = 'TRUE'). Requires the 'progress' package to be installed.
drop_outside	Whether or not to drop dates outside of 'start_date' to 'end_date' range (default = 'TRUE').
parallel	Download in parallel? Default = TRUE
ncores	Number of cores for parallel download. Default is 'parallel::detectCores()'

Value

'tibble' containing time series of NARR data for the given site

Author(s)

Alexey Shiklomanov

Examples

```
## Not run:
dat <- get_NARR_thredds("2008-01-01", "2008-01-15", 43.3724, -89.9071)

## End(Not run)
```

get_narr_url *Retrieve NARR data from a given URL*

Description

Retrieve NARR data from a given URL

Usage

```
get_narr_url(url, xy, flx, pb = NULL)
```

Arguments

url	Full URL to NARR thredds file
xy	Vector length 2 containing NARR coordinates
flx	(Logical) If 'TRUE', format for 'flx' variables. Otherwise, format for 'sfc' variables. See [narr_flx_vars].
pb	Progress bar R6 object (default = 'NULL')

Author(s)

Alexey Shiklomanov

 half_hour_downscale *half_hour_downscale*

Description

half_hour_downscale

Usage

```
temporal_downscale_half_hour(
  input_file,
  output_file,
  overwrite = TRUE,
  hr = 0.5
)
```

Arguments

input_file	location of NOAAAGEFS_1hr files
output_file	location where to store half_hour files
overwrite	whether to force hamf_hour_downscale to proceed
hr	set half hour

Value

A list of data frames is returned containing information about the data file that can be used to locate it later. Each data frame contains information about one file.

latlon2lcc	<i>Convert latitude and longitude to x-y coordinates (in km) in Lambert conformal conic projection (used by NARR)</i>
------------	---

Description

Convert latitude and longitude to x-y coordinates (in km) in Lambert conformal conic projection (used by NARR)

Usage

```
latlon2lcc(lat.in, lon.in)
```

Arguments

lat.in	Latitude coordinate
lon.in	Longitude coordinate

Value

'sp::SpatialPoints' object containing transformed x and y coordinates, in km, which should match NARR coordinates

Author(s)

Alexey Shiklomanov

latlon2narr	<i>Convert latitude and longitude coordinates to NARR indices</i>
-------------	---

Description

Convert latitude and longitude coordinates to NARR indices

Usage

```
latlon2narr(nc, lat.in, lon.in)
```

Arguments

nc	'ncdf4' connection object
lat.in	Latitude coordinate
lon.in	Longitude coordinate

Value

Vector length 2 containing NARR 'x' and 'y' indices, which can be used in 'ncdf4::ncvar_get' 'start' argument.

Author(s)

Alexey Shiklomanov

lightME	<i>Simulates the light macro environment</i>
---------	--

Description

Simulates light macro environment based on latitude, day of the year. Other coefficients can be adjusted.

Usage

```
lightME(lat = 40, DOY = 190, t.d = 12, t.sn = 12, atm.P = 1e+05, alpha = 0.85)
```

Arguments

lat	the latitude, default is 40 (Urbana, IL, U.S.).
DOY	the day of the year (1–365), default 190.
t.d	time of the day in hours (0–23), default 12.
t.sn	time of solar noon, default 12.
atm.P	atmospheric pressure, default 1e5 (kPa).
alpha	atmospheric transmittance, default 0.85.

Value

a `list` structure with components:

- 'I.dir' Direct radiation ($\mu \text{ mol } m^{-2} s^{-1}$)
- 'I.diff' Indirect (diffuse) radiation ($\mu \text{ mol } m^{-2} s^{-1}$)
- 'cos.th' cosine of θ , solar zenith angle.
- 'propIdir' proportion of direct radiation.
- 'propIdiff' proportion of indirect (diffuse) radiation.

lm_ensemble_sims	<i>lm_ensemble_sims</i>
------------------	-------------------------

Description

This function does the heavy lifting in the final function of the tdm workflow titled `predict_subdaily_met()`. It uses a linear regression approach by generating the hourly values from the coarse data of the file the user selects to downscale based on the hourly models and betas generated by `gen.subdaily.models()`.

Usage

```
lm_ensemble_sims(
  dat.mod,
  n.ens,
  path.model,
  direction.filter,
  lags.list = NULL,
  lags.init = NULL,
  dat.train,
  precip.distribution,
  force.sanity = TRUE,
  sanity.tries = 25,
  sanity.sd = 6,
  seed = Sys.time(),
  print.progress = FALSE
)
```

Arguments

`dat.mod` - dataframe to be predicted at the time step of the training data

`n.ens` - number of hourly ensemble members to generate

`path.model` - path to where the training model & betas is stored

`direction.filter`
- Whether the model will be filtered backward or forward in time. options = c("backward", "forward") (PaleON will go backward, anybody interested in the future will go forward)

`lags.init` - a data frame of initialization parameters to match the data in `dat.mod`

`dat.train` - the training data used to fit the model; needed for night/day in `surface_downwelling_shortwave_flux_in_`

`precip.distribution`
- a list with 2 sub-lists containing the number of observations with precip in the training data per day & the hour of max rain in the training data. This will be used to help solve the "constant drizzle" problem

`force.sanity` - (logical) do we force the data to meet sanity checks?

`sanity.tries` - how many time should we try to predict a reasonable value before giving up? We don't want to end up in an infinite loop

`sanity.sd` - how many standard deviations from the mean should be used to determine sane outliers (default 6)

`seed` - (optional) set the seed manually to allow reproducible results

`print.progress` - if TRUE will print progress bar

Details

Linear Regression Ensemble Simulation Met downscaling function that predicts ensembles of down-scaled meteorology

Author(s)

Christy Rollinson, James Simkins

See Also

Other tdm - Temporally Downscale Meteorology: [gen.subdaily.models\(\)](#), [model.train\(\)](#), [nc.merge\(\)](#), [predict_subdaily_met\(\)](#), [save.betas\(\)](#), [save.model\(\)](#), [subdaily_pred\(\)](#), [temporal.downscale.functions\(\)](#)

load.cfmet

Load met data from PEcAn formatted met driver

Description

subsets a PEcAn formatted met driver file and converts to a data.frame object

Usage

```
load.cfmet(met.nc, lat, lon, start.date, end.date)
```

Arguments

met.nc	object of class ncdf4 representing an open CF compliant, PEcAn standard netcdf file with met data
lat	numeric value of latitude
lon	numeric value of longitude
start.date	format is 'YYYY-MM-DD'
end.date	format is 'YYYY-MM-DD'

Value

data frame of met data

Author(s)

David LeBauer

merge_met_variable	<i>Merge a new met variable from an external file (e.g. CO2) into existing met files</i>
--------------------	--

Description

Merge a new met variable from an external file (e.g. CO2) into existing met files

Usage

```
merge_met_variable(
  in.path,
  in.prefix,
  start_date,
  end_date,
  merge.file,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

in.path	path to original data
in.prefix	prefix of original data
start_date	
end_date	
merge.file	path of file to be merged in
overwrite	logical: replace output file if it already exists?
verbose	logical: should <code>ncdf4</code> functions print debugging information as they run?
...	

Details

Currently modifies the files IN PLACE rather than creating a new copy of the files an a new DB record. Currently unit and name checking only implemented for CO2. Currently does not yet support merge data that has lat/lon New variable only has time dimension and thus MIGHT break downstream code....

Value

Currently nothing. TODO: Return a data frame summarizing the merged files.

Examples

```

## Not run:
in.path    <- "~/paleon/PaleONregional_CF_site_1-24047/"
in.prefix  <- ""
outfolder  <- "~/paleon/metTest/"
merge.file <- "~/paleon/paleon_monthly_co2.nc"
start_date <- "0850-01-01"
end_date   <- "2010-12-31"
overwrite  <- FALSE
verbose    <- TRUE

merge_met_variable(in.path,in.prefix,start_date,end_date,merge.file,overwrite,verbose)
PEcAn.DALEC::met2model.DALEC(in.path,in.prefix,outfolder,start_date,end_date)

## End(Not run)

```

met.process	<i>met.process</i>
-------------	--------------------

Description

met.process

Usage

```

met.process(
  site,
  input_met,
  start_date,
  end_date,
  model,
  host = "localhost",
  dbparms,
  dir,
  browndog = NULL,
  spin = NULL,
  overwrite = FALSE
)

```

Arguments

site	Site info from settings file
input_met	Which data source to process.
start_date	the start date of the data to be downloaded (will only use the year part of the date)
end_date	the end date of the data to be downloaded (will only use the year part of the date)
model	model_type name

host	Host info from settings file
dbparms	database settings from settings file
dir	directory to write outputs to
spin	spin-up settings passed to model-specific met2model. List containing nyear (number of years of spin-up), nsample (first n years to cycle), and resample (TRUE/FALSE)
overwrite	Whether to force met.process to proceed. 'overwrite' may be a list with individual components corresponding to 'download', 'met2cf', 'standardize', and 'met2model'. If it is instead a simple boolean, the default behavior for 'overwrite=FALSE' is to overwrite nothing, as you might expect. Note however that the default behavior for 'overwrite=TRUE' is to overwrite everything *except* raw met downloads. I.e., it corresponds to: list(download = FALSE, met2cf = TRUE, standardize = TRUE, met2model = TRUE)

Author(s)

Elizabeth Cowdery, Michael Dietze, Ankur Desai, James Simkins, Ryan Kelly

`met.process.stage` *met.process.stage*

Description

`met.process.stage`

Usage

`met.process.stage(input.id, raw.id, con)`

Arguments

<code>input.id</code>	bety db for input format
<code>raw.id</code>	format id for the raw met data
<code>con</code>	database connection

Author(s)

Elizabeth Cowdery

 met2CF.ALMA

met2CF.ALMA

Description

Get meteorology variables from ALMA netCDF files and convert to netCDF CF format

Usage

```
met2CF.ALMA(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE
)
```

Arguments

in.path	location on disk where inputs are stored
in.prefix	prefix of input and output files
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded (will only use the year part of the date)
end_date	the end date of the data to be downloaded (will only use the year part of the date)
overwrite	should existing files be overwritten

Author(s)

Mike Dietze

 met2CF.Ameriflux

met2CF.Ameriflux

Description

Get meteorology variables from Ameriflux L2 netCDF files and convert to netCDF CF format

Usage

```
met2CF.Ameriflux(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

in.path	location on disk where inputs are stored
in.prefix	prefix of input and output files
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded (will only use the year part of the date)
end_date	the end date of the data to be downloaded (will only use the year part of the date)
overwrite	should existing files be overwritten
verbose	should output of function be extra verbose

Author(s)

Josh Mantooth, Mike Dietze, Elizabeth Cowdery, Ankur Desai

met2CF.AmerifluxLBL *met2CF.AmerifluxLBL*

Description

Get meteorology variables from Ameriflux LBL and convert to netCDF CF format

Usage

```
met2CF.AmerifluxLBL(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  format,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

in.path	location on disk where inputs are stored
in.prefix	prefix of input and output files
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded (will only use the year part of the date)
end_date	the end date of the data to be downloaded (will only use the year part of the date)
format	is data frame or list with elements as described below The AmerifluxLBL format is Bety record 5000000002 which could be returned from P _E cAn.DB::query.format.vars(format.id=5000000002) format is output from db/R/query.format.vars, and should have: REQUIRED: format\$lat = latitude of site format\$lon = longitude of site format\$header = number of lines of header format\$vars is a data.frame with lists of information for each variable to read, at least airT is required format\$vars\$input_name = Name in CSV file format\$vars\$input_units = Units in CSV file format\$vars\$bety_name = Name in BETY OPTIONAL: format\$na.strings = list of missing values to convert to NA, such as -9999 format\$skip = lines to skip excluding header format\$vars\$column_number = Column number in CSV file (optional, will use header name first) Columns with NA for bety variable name are dropped. Units for datetime field are the lubridate function that will be used to parse the date (e.g. ymd_hms or mdy_hm).
overwrite	should existing files be overwritten
verbose	should output of function be extra verbose

Author(s)

Ankur Desai

met2CF.csv*Convert met data from CSV to CF*

Description

Convert met data from CSV to CF

Usage

```
met2CF.csv(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  format,
  lat = NULL,
```

```

lon = NULL,
nc_verbose = FALSE,
overwrite = FALSE,
...
)

```

Arguments

<code>in.path</code>	directory in which to find met csv files
<code>in.prefix</code>	pattern to match to find met files inside 'in.path'
<code>outfolder</code>	directory name to write CF outputs
<code>start_date, end_date</code>	when to start and stop conversion. Specify as 'Date' objects, but only the year component is used
<code>format</code>	data frame or list produced by 'PEcAn.DB::query.format.vars'. See details
<code>lat, lon</code>	latitude and longitude of site, in decimal degrees. If not provided, these are taken from 'format'.
<code>nc_verbose</code>	logical: run nvar_add in verbose mode?
<code>overwrite</code>	Logical: Redo conversion if output file already exists?
<code>...</code>	other arguments, currently ignored

Details

The 'format' argument takes an output from 'PEcAn.DB::query.format.vars', and should have the following components:

REQUIRED:

- 'format\$lat': latitude of site (unless passed by 'lat')
- 'format\$lon': longitude of site (unless passed by 'lon')
- 'format\$header': number of lines of header
- 'format\$vars': a data.frame with lists of information for each variable to read. At least 'airT' is required
- 'format\$vars\$input_name': name in CSV file
- 'format\$vars\$input_units': units in CSV file
- 'format\$vars\$bety_name': name in BETY. See https://pecan.gitbooks.io/pecan-documentation/content/developers_guide/an-Input-Converter.html for allowable names.

OPTIONAL:

- 'format\$na.strings': list of missing values to convert to NA, such as -9999
- 'format\$skip': lines to skip excluding header
- 'format\$vars\$column_number': column number in CSV file (optional, will use header name first)

Columns with NA for bety variable name are dropped.

Units for datetime field are the lubridate function that will be used to parse the date (e.g. ymd_hms or mdy_hm).

Author(s)

Mike Dietze, David LeBauer, Ankur Desai

Examples

```
## Not run:
con <- PEEAn.DB::db.open(
  list(user='bety', password='bety', host='localhost',
        dbname='bety', driver='PostgreSQL',write=TRUE))
start_date <- lubridate::ymd_hm('200401010000')
end_date <- lubridate::ymd_hm('200412312330')
file<-PEEAn.data.atmosphere::download.Fluxnet2015('US-WCr', '~/ ', start_date, end_date)
in.path <- '~/ '
in.prefix <- file$dbfile.name
outfolder <- '~/ '
format.id <- 5000000001
format <- PEEAn.DB::query.format.vars(format.id=format.id,bety = bety)
format$lon <- -92.0
format$lat <- 45.0
format$time_zone <- "America/Chicago"
results <- PEEAn.data.atmosphere::met2CF.csv(
  in.path, in.prefix, outfolder,
  start_date, end_date, format,
  overwrite=TRUE)

## End(Not run)
```

met2CF.ERA5

met2cf.ERA5

Description

met2cf.ERA5

Usage

```
met2CF.ERA5(
  lat,
  long,
  start_date,
  end_date,
  sitename,
  outfolder,
  out.xts,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

lat	latitude
long	longitude
start_date	start date
end_date	end date
sitename	The name of the site used for making the identifier.
outfolder	Path to directory where nc files need to be saved.
out.xts	Output of the extract.nc.ERA5 function which is a list of time series of met variables for each ensemble member.
overwrite	Logical if files needs to be overwritten.
verbose	Logical flag defining if ouput of function be extra verbose.

Value

list of dataframes

met2CF.FACE	<i>convert FACE files to CF files</i>
-------------	---------------------------------------

Description

Note: 'in.path' and 'in.prefix' together must identify exactly one file, or this function returns NULL. Further note that despite its name, 'in.prefix' will match anywhere in the filename: met2CF.FACE("dir", "a", ...) will find both 'dir/a_b.nc' and 'dir/b_a.nc'!

Usage

```
met2CF.FACE(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  input.id,
  site,
  format,
  ...
)
```

Arguments

<code>in.path</code>	directory in which to find inputs (as <code>*.nc</code>)
<code>in.prefix</code>	pattern to match to select a file within <code>in.path</code>
<code>outfolder</code>	path to write output. Should contain the substring "FACE", which will be rewritten to "FACE_a" and "FACE_e" for the corresponding treatments.
<code>start_date, end_date</code>	ignored. Time is taken from the input files.
<code>input.id</code>	ignored
<code>site</code>	list[like]. Only components <code>'lat'</code> and <code>'lon'</code> (both in decimal degrees) are currently used
<code>format</code>	specification of variable names and units in the format returned by <code>'PEcAn.DB::query.format.vars'</code>
<code>...</code>	other arguments, currently ignored

Author(s)

Elizabeth Cowdery

`met2CF.Geostreams` *Convert geostreams JSON to CF met file*

Description

Convert geostreams JSON to CF met file

Usage

```
met2CF.Geostreams(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

<code>in.path</code>	directory containing Geostreams JSON file(s) to be converted
<code>in.prefix</code>	initial portion of input filenames (everything before the dates)
<code>outfolder</code>	directory where nc output files should be written. Will be created if necessary
<code>start_date, end_date</code>	beginning and end of run, YYYY-MM-DD.

overwrite	logical: Regenerate existing files of the same name?
verbose	logical, passed on to <code>nc_create</code> to control how chatty it should be during netCDF creation
...	other arguments, currently ignored

Author(s)

Harsh Agrawal, Chris Black

met2CF.ICOS

Convert variables ICOS variables to CF format.

Description

Variables present in the output netCDF file: `air_temperature`, `air_temperature`, `relative_humidity`, `specific_humidity`, `water_vapor_saturation_deficit`, `surface_downwelling_longwave_flux_in_air`, `surface_downwelling_shortwave_flux_in_air`, `surface_downwelling_photosynthetic_photon_flux_in_air`, `precipitation_flux`, `eastward_wind`, `northward_wind`

Usage

```
met2CF.ICOS(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  format,
  overwrite = FALSE,
  ...
)
```

Arguments

<code>in.path</code>	path to the input ICOS product CSV file
<code>in.prefix</code>	name of the input file
<code>outfolder</code>	path to the directory where the output file is stored. If specified directory does not exist, it is created.
<code>start_date</code>	start date of the input file
<code>end_date</code>	end date of the input file
<code>format</code>	format is data frame or list with elements as described below REQUIRED: <code>format\$header</code> = number of lines of header <code>format\$vars</code> is a data.frame with lists of information for each variable to read, at least <code>airT</code> is required <code>format\$vars\$input_name</code> = Name in CSV file <code>format\$vars\$input_units</code> = Units in CSV file <code>format\$vars\$bety_name</code>

= Name in BETY OPTIONAL: format\$lat = latitude of site format\$lon = longitude of site format\$na.strings = list of missing values to convert to NA, such as -9999 format\$skip = lines to skip excluding header format\$vars\$column_number = Column number in CSV file (optional, will use header name first) Columns with NA for bety variable name are dropped.

overwrite overwrite should existing files be overwritten. Default False.

... used when extra arguments are present.

Value

information about the output file

met2CF.NARR	<i>Convert NARR files to CF files</i>
-------------	---------------------------------------

Description

Convert NARR files to CF files

Usage

```
met2CF.NARR(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

in.path	directory in which to find NARR files
in.prefix	pattern to match to find NARR files inside 'in.path'
outfolder	directory name to write CF outputs
start_date	the start date of the data to be downloaded (will only use the year part of the date)
end_date	the end date of the data to be downloaded (will only use the year part of the date)
overwrite	should existing files be overwritten
verbose	should output of function be extra verbose
...	other arguments, currently ignored

Author(s)

Elizabeth Cowdery, Rob Kooper

met2CF.PaleON	<i>met2CF.PaleON</i>
---------------	----------------------

Description

Get meteorology variables from PaleON netCDF files and convert to netCDF CF format

Usage

```
met2CF.PaleON(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  lat,
  lon,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

in.path	location on disk where inputs are stored
in.prefix	prefix of input and output files
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded (will only use the year part of the date)
end_date	the end date of the data to be downloaded (will only use the year part of the date)
overwrite	should existing files be overwritten

Author(s)

Mike Dietze

met2CF.PaleONregional	<i>met2CF.PaleONregional</i>
-----------------------	------------------------------

Description

Get meteorology variables from PaleON netCDF files and convert to netCDF CF format

Usage

```
met2CF.PaleONregional(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

in.path	location on disk where inputs are stored
in.prefix	prefix of input and output files
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded (will only use the year part of the date)
end_date	the end date of the data to be downloaded (will only use the year part of the date)
overwrite	should existing files be overwritten

Author(s)

Mike Dietze

metgapfill	<i>Take an Ameriflux NetCDF file Fill missing met values using MDS approach using MPI-BGC REddyProc library Currently Future version: Choose which variables to gap fill Future version will first downscale and fill with NARR, then REddyProc</i>
------------	---

Description

Take an Ameriflux NetCDF file Fill missing met values using MDS approach using MPI-BGC REddyProc library Currently Future version: Choose which variables to gap fill Future version will first downscale and fill with NARR, then REddyProc

Usage

```
metgapfill(
  in.path,
  in.prefix,
  outfolder,
  start_date,
```

```

    end_date,
    lst = 0,
    overwrite = FALSE,
    verbose = FALSE,
    ...
)

```

Arguments

in.path	location on disk where inputs are stored
in.prefix	prefix of input and output files
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be downloaded (will only use the year part of the date)
end_date	the end date of the data to be downloaded (will only use the year part of the date)
lst	is timezone offset from UTC, if timezone is available in time:units attribute in file, it will use that, default is to assume UTC
overwrite	should existing files be overwritten
verbose	should the function be very verbose

Author(s)

Ankur Desai

metgapfill.NOAA_GEFS *Gapfill NOAA_GEFS weather data*

Description

Gapfill NOAA_GEFS weather data

Usage

```

metgapfill.NOAA_GEFS(
  in.prefix,
  in.path,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)

```


Arguments

in.prefix	the met file name
in.path	The location of the file
outfolder	The place to write the output file to
start_date	The start date of the contents of the file
end_date	The end date of the contents of the file
overwrite	Whether or not to overwrite the output file if it exists or not
verbose	Passed to nc writing functions for additional output

Purpose

This function uses simple methods to gapfill NOAA GEFS met data Temperature and Precipitation are gapfilled with spline; other data sources are gapfilled with using linear models fitted to other fitted data.

Author(s)

Luke Dramko

met_temporal_downscale.Gaussian_ensemble
met_temporal_downscale.Gaussian_ensemble

Description

takes source data and a training dataset from the same site and temporally downscales the source dataset to the resolution of the training dataset based on statistics of the training dataset.

Usage

```
met_temporal_downscale.Gaussian_ensemble(  
    in.path,  
    in.prefix,  
    outfolder,  
    input_met,  
    train_met,  
    overwrite = FALSE,  
    verbose = FALSE,  
    swdn_method = "sine",  
    n_ens = 10,  
    w_len = 20,  
    utc_diff = -6,  
    ...  
)
```

Arguments

in.path	ignored
in.prefix	ignored
outfolder	path to directory in which to store output. Will be created if it does not exist
input_met	- the source dataset that will temporally downscaled by the train_met dataset
train_met	- the observed dataset that will be used to train the modeled dataset in NC format. i.e. Flux Tower dataset (see download.Fluxnet2015 or download.Ameriflux)
overwrite	logical: replace output file if it already exists?
verbose	logical: should <code>ncdf4</code> functions print debugging information as they run?
swdn_method	- Downwelling shortwave flux in air downscaling method (options are "sine", "spline", and "Waichler")
n_ens	- numeric value with the number of ensembles to run
w_len	- numeric value that is the window length in days
utc_diff	- numeric value in HOURS that is local standard time difference from UTC time. CST is -6
...	further arguments, currently ignored

Author(s)

James Simkins

model.train

model.train

Description

Function to create linear regression models for specific met variables. This is used in conjunction with `temporal.downscale.functions()` to generate linear regression statistics and save their output to be called later in `lm_ensemble_sims()`.

Usage

```
model.train(dat.subset, v, n.beta, resids = resids, threshold = NULL, ...)
```

Arguments

dat.subset	data.frame containing lags, next, and downscale period data
n.beta	number of betas to pull from
resids	TRUE or FALSE, whether to use residuals or not
threshold	NULL except for <code>surface_downwelling_shortwave_radiation</code> , helps with our distinction between day and night (no shortwave without sunlight)

Details

TDM Model Train Linear regression calculations for specific met variables

Author(s)

Christy Rollinson, James Simkins

See Also

Other tdm - Temporally Downscale Meteorology: [gen.subdaily.models\(\)](#), [lm_ensemble_sims\(\)](#), [nc.merge\(\)](#), [predict_subdaily_met\(\)](#), [save.betas\(\)](#), [save.model\(\)](#), [subdaily_pred\(\)](#), [temporal.downscale.function\(\)](#)

narr_flux_vars	<i>NARR flux and sfc variables</i>
----------------	------------------------------------

Description

NARR flux and sfc variables

Usage

narr_flux_vars

narr_sfc_vars

narr_all_vars

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 5 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 3 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 8 rows and 3 columns.

<code>nc.merge</code>	<i>nc.merge</i>
-----------------------	-----------------

Description

This is the 1st function for the tdm (Temporally Downscale Meteorology) workflow. The `nc2dat.train` function parses multiple netCDF files into one central training data file called 'dat.train_file'. This netCDF file will be used to generate the subdaily models in the next step of the workflow, `generate.subdaily.models()`. It is also called in `tdm_predict_subdaily_met` which is the final step of the tdm workflow.

Usage

```
nc.merge(  
  outfolder,  
  in.path,  
  in.prefix,  
  start_date,  
  end_date,  
  upscale = FALSE,  
  overwrite = FALSE,  
  verbose = FALSE,  
  ...  
)
```

Arguments

outfolder	- directory where output will be stored
in.path	- path of coarse model (e.g. GCM output)
in.prefix	- prefix of model string as character (e.g. IPSL.r1i1p1.rcp85)
start_date	- yyyy-mm-dd
end_date	- yyyy-mm-dd
upscale	- Upscale can either be set for FALSE (leave alone) or to the temporal resolution you want to aggregate to
overwrite	logical: replace output file if it already exists?
verbose	logical: should ncdf4 functions print debugging information as they run?

Details

nc.merge Parses multiple netCDF files into one central document for temporal downscaling procedure

Author(s)

James Simkins, Christy Rollinson

See Also

Other tdm - Temporally Downscale Meteorology: [gen.subdaily.models\(\)](#), [lm_ensemble_sims\(\)](#), [model.train\(\)](#), [predict_subdaily_met\(\)](#), [save.betas\(\)](#), [save.model\(\)](#), [subdaily_pred\(\)](#), [temporal.downscale.functions\(\)](#)

noaa_grid_download	<i>Download gridded forecast in the box bounded by the latitude and longitude list</i>
--------------------	--

Description

Download gridded forecast in the box bounded by the latitude and longitude list

Usage

```
noaa_grid_download(  
  lat_list,  
  lon_list,  
  forecast_time,  
  forecast_date,  
  model_name_raw,  
  output_directory,  
  end_hr  
)
```

Arguments

lat_list	lat for site
lon_list	long for site
forecast_time	start hour of forecast
forecast_date	date for forecast
model_name_raw	model name for directory creation
output_directory	output directory
end_hr	end hr to determine how many hours to download

Value

NA

noaa_stage2	<i>noaa_stage2</i>
-------------	--------------------

Description

noaa_stage2

Usage

```
noaa_stage2(
  cycle = 0,
  version = "v12",
  endpoint = "data.ecoforecast.org",
  verbose = TRUE,
  start_date = ""
)
```

Arguments

cycle	Hour at which forecast was made, as character string ("00", "06", "12" or "18"). Only "00" (default) has 30 days horizon.
version	GEFS forecast version. Prior versions correspond to forecasts issued before 2020-09-25 which have different ensemble number and horizon, among other changes, and are not made available here. Leave as default.
endpoint	the EFI host address (leave as default)
verbose	logical, displays or hides messages
start_date	forecast start date yyyy-mm-dd format

Author(s)

Alexis Helgeson (taken from neon4cast package)

par2ppfd

par2ppfd

Description

convert PAR to PPFd

Usage

```
par2ppfd(watts)
```

Arguments

watts	PAR (W / m2)
-------	--------------

Details

Converts photosynthetically active radiation (PAR, units of Watts / m2) to photosynthetic photon flux density (PPFD) in units of umol / m2 / s From Campbell and Norman p151 $PPFD = PAR * (J/m2/s) * (1 \text{ mol} / 2.35e5 \text{ J})$ 2.35e5 J / mol is the energy content of solar radiation in the PAR waveband

Value

PPFD (umol / m2 / s)

Author(s)

David LeBauer

pecan_standard_met_table

Conversion table for PEcAn standard meteorology

Description

Conversion table for PEcAn standard meteorology

Usage

pecan_standard_met_table

Format

An object of class tbl_df (inherits from tbl, data.frame) with 18 rows and 8 columns.

permute.nc

permute.nc

Description

Permute netCDF files

Usage

```
permute.nc(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

in.path	location on disk where inputs are stored
in.prefix	prefix of input and output files
outfolder	location on disk where outputs will be stored
start_date	the start date of the data to be permuted (will only use the year part of the date)
end_date	the end date of the data to be permuted (will only use the year part of the date)
overwrite	should existing files be overwritten
verbose	should output of function be extra verbose

Author(s)

Elizabeth Cowdery, Rob Kooper

post_process	<i>Post process raw NARR downloaded data frame</i>
--------------	--

Description

Post process raw NARR downloaded data frame

Usage

```
post_process(dat)
```

Arguments

dat	Nested 'tibble' from mapped call to [get_narr_url]
-----	--

predict_subdaily_met	<i>predict_subdaily_met</i>
----------------------	-----------------------------

Description

This is the main function of the tdm family workflow. This function predicts subdaily meteorology from daily means using a linear regression modeling approach. It takes a dataset with daily resolution and temporally downscales it to hourly resolution using the statistics generated by `gen.subdaily.models()`. It references the `predict.subdaily.function` located in `lm_ensemble_sims()` which uses a linear regression based approach to downscale. We generate multiple ensembles of possible hourly values dictated from the models and betas generated in `gen.subdaily.models`. Each ensemble member is saved as a netCDF file in CF conventions and these files are ready to be used in the general PEcAn workflow.

Usage

```

predict_subdaily_met(
  outfolder,
  in.path,
  in.prefix,
  path.train,
  direction.filter = "forward",
  lm.models.base,
  yrs.predict = NULL,
  ens.labs = 1:3,
  resids = FALSE,
  adjust.pr = 1,
  force.sanity = TRUE,
  sanity.tries = 25,
  overwrite = FALSE,
  verbose = FALSE,
  seed = format(Sys.time(), "%m%d"),
  print.progress = FALSE,
  ...
)

```

Arguments

outfolder	- directory where output file will be stored
in.path	- base path to dataset you wish to temporally downscale; Note: in order for parallelization to work, the in.prefix will need to be appended as the final level of the file structure. For example, if prefix is GFDL.CM3.rcp45.r1i1p1, there should be a directory with that title in in.path.
in.prefix	- prefix of model dataset, i.e. if file is GFDL.CM3.rcp45.r1i1p1.2006 the prefix is 'GFDL.CM3.rcp45.r1i1p1'
path.train	- path to CF/PEcAn style training data where each year is in a separate file.
direction.filter	- Whether the model will be filtered backward or forwards in time. options = c("backward", "forwards") (default is forward; PaleON will go backward, anybody interested in the future will go forwards)
lm.models.base	- path to linear regression model folders generated using gen.subdaily.models
yrs.predict	- years for which you want to generate met. if NULL, all years in in.path will be done
ens.labs	- vector containing the labels (suffixes) for each ensemble member; this allows you to add to your ensemble rather than overwriting with a default naming scheme
resids	- logical stating whether to pass on residual data or not
adjust.pr	- adjustment factor fore precipitation when the extracted values seem off
force.sanity	- (logical) do we force the data to meet sanity checks?

sanity.tries - how many time should we try to predict a reasonable value before giving up?
We don't want to end up in an infinite loop

overwrite logical: replace output file if it already exists?

verbose logical: should `ncdf4` functions print debugging information as they run?

seed - manually set seed for results to be reproducible

print.progress - print the progress bar?

Details

Predict Subdaily Meteorology Predict Subdaily Meteorology based off of statistics created in `gen.subdaily.models()`

Author(s)

Christy Rollinson, James Simkins

See Also

Other tdm - Temporally Downscale Meteorology: [gen.subdaily.models\(\)](#), [lm_ensemble_sims\(\)](#), [model.train\(\)](#), [nc.merge\(\)](#), [save.betas\(\)](#), [save.model\(\)](#), [subdaily_pred\(\)](#), [temporal.downscale.functions\(\)](#)

Examples

```
## Not run:
library(PEcAn.data.atmosphere)
outfolder = '~/Downscaled_GCM'
in.path = '~/raw_GCM'
in.prefix = 'GFDL'
lm.models.base = 'sf_scratch/US-WCr'
dat.train_file = 'Training_data/US-WCr_dat.train.nc'
start_date = '2010-01-01'
end_date = '2014-12-31'
cores.max = 12
n.ens = 3
## End(Not run)
```

prepare_narr_year *Write NetCDF file for a single year of data*

Description

Write NetCDF file for a single year of data

Usage

```
prepare_narr_year(dat, file, lat_nc, lon_nc, verbose = FALSE)
```

Arguments

dat	NARR tabular data for a single year ([get_NARR_thredds])
file	Full path to target file
lat_nc	'ncdim' object for latitude
lon_nc	'ncdim' object for longitude
verbose	logical: ask 'ncdf4' functions to be very chatty while they work?

Value

List of NetCDF variables in data. Creates NetCDF file containing data as a side effect

process_gridded_noaa_download

Extract and temporally downscale points from downloaded grid files

Description

Extract and temporally downscale points from downloaded grid files

Usage

```
process_gridded_noaa_download(
  lat_list,
  lon_list,
  site_id,
  downscale,
  overwrite,
  forecast_date,
  forecast_time,
  model_name,
  model_name_ds,
  model_name_raw,
  output_directory
)
```

Arguments

lat_list	lat for site
lon_list	lon for site
site_id	Unique site_id for file creation
downscale	Logical. Default is TRUE. Downscales from 6hr to hourly
overwrite	Logical. Default is FALSE. Should existing files be overwritten
forecast_date	Date for download
forecast_time	Time (0,6,12,18) for start of download

model_name Name of model for file name
 model_name_ds Name of downscale file name
 model_name_raw Name of raw file name
 output_directory
 Output directory

Value

List

qair2rh	<i>qair2rh</i>
---------	----------------

Description

Convert specific humidity to relative humidity

Usage

```
qair2rh(qair, temp, press = 1013.25)
```

Arguments

qair specific humidity, dimensionless (e.g. kg/kg) ratio of water mass / total air mass
 temp degrees C
 press pressure in mb

Details

converting specific humidity into relative humidity NCEP surface flux data does not have RH from Bolton 1980 Teh computation of Equivalent Potential Temperature https://archive.eol.ucar.edu/projects/ceop/dm/documents/refdata_report/eqns.html

Value

rh relative humidity, ratio of actual water mixing ratio to saturation mixing ratio

Author(s)

David LeBauer

read.register	<i>read.register</i>
---------------	----------------------

Description

read.register

Usage

read.register(register.xml, con)

Arguments

register.xml	path of xml file
con	betydb connection

Author(s)

Betsy Cowdery

read_narr_var	<i>Read a specific variable from a NARR NetCDF file</i>
---------------	---

Description

Read a specific variable from a NARR NetCDF file

Usage

read_narr_var(nc, xy, variable, unit, flx, pb = NULL)

Arguments

nc	'ncdf4' connection object
xy	Vector length 2 containing NARR coordinates
variable	NARR name of variable to retrieve
unit	Output unit of variable to retrieve
flx	(Logical) If 'TRUE', format for 'flx' variables. Otherwise, format for 'sfc' variables. See [narr_flx_vars].
pb	Progress bar R6 object (default = 'NULL')

Author(s)

Alexey Shiklomanov

rh2qair *RH to SH*

Description

converts relative humidity to specific humidity

Usage

```
rh2qair(rh, T, press = 101325)
```

Arguments

rh	relative humidity (proportion, not %)
T	absolute temperature (Kelvin)
press	air pressure (Pascals)

Author(s)

Mike Dietze, Ankur Desai

save.betas *save.betas*

Description

Function to save betas as a .nc file. This is utilized in gen.subdaily.models() when linear regression models are created

Usage

```
save.betas(model.out, betas, outfile)
```

Arguments

model.out	list linear regression model output
betas	name of the layer of betas to save (e.g. 'betas' or 'betas.resid')
outfile	location where output will be stored

Details

TDM Save Betas Saves betas that are calculated during gen.subdaily.models()

Author(s)

Christy Rollinson, James Simkins

See Also

Other tdm - Temporally Downscale Meteorology: [gen.subdaily.models\(\)](#), [lm_ensemble_sims\(\)](#), [model.train\(\)](#), [nc.merge\(\)](#), [predict_subdaily_met\(\)](#), [save.model\(\)](#), [subdaily_pred\(\)](#), [temporal.downscale.fun](#)

 save.model

save.model

Description

Function to save models as a .nc file. This is utilized in `gen.subdaily.models()` when linear regression models are created

Usage

```
save.model(model.out, model, outfile)
```

Arguments

model.out	list linear regression model output
model	name of the layer of model to save (e.g. 'model' or 'model.resid')
outfile	location where output will be stored

Details

TDM Save Models Saves models that are created during `gen.subdaily.models()`

Author(s)

Christy Rollinson, James Simkins

See Also

Other tdm - Temporally Downscale Meteorology: [gen.subdaily.models\(\)](#), [lm_ensemble_sims\(\)](#), [model.train\(\)](#), [nc.merge\(\)](#), [predict_subdaily_met\(\)](#), [save.betas\(\)](#), [subdaily_pred\(\)](#), [temporal.downscale.fun](#)

site.lst	<i>Find time zone for a site</i>
----------	----------------------------------

Description

Find time zone for a site

Usage

```
site.lst(site.id, con)
```

Arguments

site.id	bety id of site to look up
con	betydb connection object

Author(s)

Betsy Cowdery

site_from_tag	<i>Function to find the site code for a specific tag</i>
---------------	--

Description

Example: sitename = 'Rhinelander Aspen FACE Experiment (FACE-RHIN)' tag = 'FACE' site_from_tag(sitename,tag) = 'RHIN' Requires that site names be set up specifically with (tag-sitecode) - this may change

Usage

```
site_from_tag(sitename, tag)
```

Arguments

sitename	full name of site
tag	abbreviated name of site

Author(s)

Betsy Cowdery

solarMJ2ppfd	<i>MJ to PPFd</i>
--------------	-------------------

Description

Solar Radiation to PPFd

Usage

solarMJ2ppfd(solarMJ)

Arguments

solarMJ MJ per day

Details

There is no easy straight way to convert MJ/m² to $\mu\text{mol photons} / \text{m}^2 / \text{s}$ (PAR). Note: 1 Watt = 1J/s The above conversion is based on the following reasoning 0.12 is about how much of the total radiation is expected to occur during the hour of maximum insolation (it is a guesstimate) 2.07 is a coefficient which converts from MJ to mol photons (it is approximate and it is taken from ... Campbell and Norman (1998). Introduction to Environmental Biophysics. pg 151 'the energy content of solar radiation in the PAR waveband is $2.35 \times 10^5 \text{ J/mol}$ ' See also the chapter radiation basics (10) Here the input is the total solar radiation so to obtain in the PAR spectrum need to multiply by 0.486 This last value 0.486 is based on the approximation that PAR is 0.45-0.50 of the total radiation This means that $1\text{e}6 / (2.35\text{e}6) * 0.486 = 2.07 \text{e}6$ converts from mol to μmol 1/3600 divides the values in hours to seconds

ValuePPFD $\mu\text{mol} / \text{m}^2 / \text{s}$ **Author(s)**

Fernando Miguez

David LeBauer

spin.met

*Spin-up meteorology***Description**

Spin-up meteorology

Usage

```
spin.met(
  in.path,
  in.prefix,
  start_date,
  end_date,
  nyear = 1000,
  nsample = 50,
  resample = TRUE,
  run_start_date = start_date,
  overwrite = TRUE
)
```

Arguments

in.path	met input folder path
in.prefix	met input file prefix (shared by all annual files, can be "")
start_date	start of met
end_date	end of met
nyear	number of years of spin-up, default 1000
nsample	sample the first nsample years of met, default 50
resample	resample (TRUE, default) or cycle (FALSE) meteorology
run_start_date	date the run itself starts, which can be different than the start of met
overwrite	whether to replace previous resampling

Details

spin.met works by creating symbolic links to the sampled met file, rather than copying the whole file. Be aware that the internal dates in those files are not modified. Right now this is designed to be called within met2model.[MODEL] before the met is processed (it's designed to work with annual CF files not model-specific files) for example with models that process met into one large file

Value

updated start date

Examples

```

start_date <- "0850-01-01 00:00:00"
end_date   <- "2010-12-31 23:59:59"
nyear      <- 10
nsample    <- 50
resample   <- TRUE

## Not run:
if(!is.null(spin)){
  ## if spinning up, extend processed met by resampling or cycling met
  start_date <- PEcAn.data.atmosphere::spin.met(
    in.path, in.prefix,
    start_date, end_date,
    nyear, nsample, resample)
}

## End(Not run)

```

split_wind

*Split wind_speed into eastward_wind and northward_wind***Description**

Split wind_speed into eastward_wind and northward_wind

Usage

```

split_wind(
  in.path,
  in.prefix,
  start_date,
  end_date,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)

```

Arguments

in.path	path to original data
in.prefix	prefix of original data
start_date	
end_date	
overwrite	logical: replace output file if it already exists?
verbose	logical: should <code>ncdf4</code> functions print debugging information as they run?
...	other arguments, currently ignored

Details

Currently modifies the files IN PLACE rather than creating a new copy of the files an a new DB record.

Value

nothing. TODO: Return data frame summarizing results

Examples

```
## Not run:
in.path   <- "~/paleon/PaleONregional_CF_site_1-24047/"
in.prefix <- ""
outfolder <- "~/paleon/metTest/"
start_date <- "0850-01-01"
end_date   <- "2010-12-31"
overwrite  <- FALSE
verbose    <- TRUE

split_wind(in.path, in.prefix, start_date, end_date, merge.file, overwrite, verbose)

## End(Not run)
```

step_means

take mean at fixed intervals along a vector

Description

take mean at fixed intervals along a vector

Usage

```
step_means(x, step)
```

Arguments

x	numeric vector
step	integer step size

Details

User should check that `length(x)` is an even multiple of `step`

Value

numeric of length `length(x)/step`

subdaily_pred	<i>subdaily_pred</i>
---------------	----------------------

Description

Function to pull objects created in linear regression models and are used to predict subdaily meteorology. This function is called in `lm_ensemble_sims()` to downscale a meteorology product. Linear regression models are created in `gen.subdaily.models()`

Usage

```
subdaily_pred(
  newdata,
  model.predict,
  Rbeta,
  resid.err = FALSE,
  model.resid = NULL,
  Rbeta.resid = NULL,
  n.ens
)
```

Arguments

<code>newdata</code>	dataframe with data to be downscaled
<code>model.predict</code>	saved linear regression model
<code>Rbeta</code>	matrix with Rbetas from saved linear regression model
<code>resid.err</code>	logical, whether to include residual error or not
<code>model.resid</code>	data.frame of model residuals
<code>Rbeta.resid</code>	data.frame of Rbeta residuals
<code>n.ens</code>	number of ensembles to create

Details

Subdaily Prediction Pulls information from linear regression models to predict subdaily meteorology

Author(s)

Christy Rollinson, James Simkins

See Also

Other tdm - Temporally Downscale Meteorology: [gen.subdaily.models\(\)](#), [lm_ensemble_sims\(\)](#), [model.train\(\)](#), [nc.merge\(\)](#), [predict_subdaily_met\(\)](#), [save.betas\(\)](#), [save.model\(\)](#), [temporal.downscale.function\(\)](#)

sw2par	<i>SW to PAR</i>
--------	------------------

Description

Solar Radiation to PPFD

Usage

sw2par(sw)

Arguments

sw shortwave radiation (W/m2 == J/m2/s)

Details

Here the input is the total solar radiation so to obtain in the PAR spectrum need to multiply by 0.486 From Campbell and Norman p151 This is based on the approximation that PAR is 0.45-0.50 of the total radiation

Value

PAR W/m2

Author(s)

David LeBauer

sw2ppfd	<i>SW to PPFD</i>
---------	-------------------

Description

CF Shortwave to PPFD

Usage

sw2ppfd(sw)

Arguments

sw CF surface_downwelling_shortwave_flux_in_air (W/m2)

Details

Cambell and Norman 1998 p 151, ch 10

ValuePPFD $\mu\text{mol} / \text{m}^2 / \text{s}$ **Author(s)**

David LeBauer

`temporal.downscale.functions`*temporal_downscale_functions*

Description

This function contains the functions that do the heavy lifting in `gen.subdaily.models()` and `predict.subdaily.workflow()`. Individual variable functions actually generate the models and betas from the `dat.train_file` and save them in the output file. `save.model()` and `save.betas()` are helper functions that save the linear regression model output to a specific location. In the future, we should only save the data that we actually use from the linear regression model because this is a large file. `predict.met()` is called from `predict.subdaily.workflow()` and references the linear regression model output to predict the ensemble data.

Usage

```
temporal.downscale.functions(  
  dat.train,  
  n.beta,  
  day.window,  
  resids = FALSE,  
  parallel = FALSE,  
  n.cores = NULL,  
  seed = format(Sys.time(), "%m%d"),  
  outfolder,  
  print.progress = FALSE,  
  ...  
)
```

Arguments

<code>dat.train</code>	- training data generated by <code>tdm_nc2dat.train.R</code>
<code>n.beta</code>	- number of betas to generate
<code>day.window</code>	- number of days surrounding current day we want to pull statistics from
<code>resids</code>	- whether or not to propagate residuals, set to <code>FALSE</code>
<code>parallel</code>	- whether or not to run in parallel. this is a feature still being worked on, set to <code>FALSE</code>
<code>n.cores</code>	- number of cores to use parallel processing on, set to <code>NULL</code>

seed - allows this to be reproducible
 outfolder = where the output should be stored
 print.progress - print progress of model generation?

Details

Temporal Downscale Functions Met variable functions that are called in `gen.subdaily.models` and `predict.subdaily.workflow`

Author(s)

Christy Rollinson, James Simkins

See Also

Other tdm - Temporally Downscale Meteorology: `gen.subdaily.models()`, `lm_ensemble_sims()`, `model.train()`, `nc.merge()`, `predict_subdaily_met()`, `save.betas()`, `save.model()`, `subdaily_pred()`

temporal_downscale *Downscale NOAA GEFS from 6hr to 1hr*

Description

Downscale NOAA GEFS from 6hr to 1hr

Usage

```
temporal_downscale(input_file, output_file, overwrite = TRUE, hr = 1)
```

Arguments

input_file full path to 6hr file
 output_file full path to 1hr file that will be generated
 overwrite logical stating to overwrite any existing output_file
 hr time step in hours of temporal downscaling (default = 1)

Value

None

Author(s)

Quinn Thomas

upscale_met	<i>upscale_met</i>
-------------	--------------------

Description

upscale_met upscales the temporal resolution of a dataset

Usage

```
upscale_met(
  outfolder,
  input_met,
  resolution = 1/24,
  overwrite = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

outfolder	path to directory where output should be saved Output is netcdf named as <input_met_filename>.upscaled.nc
input_met	path to netcdf file containing met dataset
resolution	desired output resolution, in days
overwrite	logical: replace output file if it already exists?
verbose	logical: should <code>ncdf4</code> functions print debugging information as they run?
...	other arguments, currently ignored

Author(s)

James Simkins, Chris Black

wide2long	<i>Wide to Long</i>
-----------	---------------------

Description

Convert raster to lat, lon, var

Usage

```
wide2long(data.wide, lat, lon, var)
```

Arguments

data.wide	data
lat	latitude for rows
lon	longitude for columns
var	variable being measured

Value

data.frame with colnames (lat, lon, var)

Author(s)

David LeBauer

write_noaa_gefs_netcdf

Write NOAA GEFS netCDF

Description

Write NOAA GEFS netCDF

Usage

```
write_noaa_gefs_netcdf(
  df,
  ens = NA,
  lat,
  lon,
  cf_units,
  output_file,
  overwrite
)
```

Arguments

df	data frame of meteorological variables to be written to netcdf. Columns must start with time with the following columns in the order of 'cf_units'
ens	ensemble index used for subsetting df
lat	latitude in degree north
lon	longitude in degree east
cf_units	vector of variable names in order they appear in df
output_file	name, with full path, of the netcdf file that is generated
overwrite	logical to overwrite existing netcdf file

Value

NA

Author(s)

Quinn Thomas

Index

- * **datasets**
 - narr_flx_vars, 83
 - pecan_standard_met_table, 87
- * **debias - Debias & Align Meteorology Datasets into continuous time series**
 - align.met, 5
 - debias.met.regression, 13
- * **models**
 - lightME, 63
- * **tdm - Temporally Downscale Meteorology**
 - gen.subdaily.models, 53
 - lm_ensemble_sims, 63
 - model.train, 82
 - nc.merge, 83
 - predict_subdaily_met, 88
 - save.betas, 94
 - save.model, 95
 - subdaily_pred, 101
 - temporal.downscale.functions, 103
- .download.raw.met.module
 - (download.raw.met.module), 38
- AirDens, 4
- align.met, 5, 15
- browndog.met, 6
- build_cf_variables_table_url, 7, 59
- cfmet.downscale.daily, 8
- cfmet.downscale.subdaily, 9
- cfmet.downscale.time, 9
- check_met_input_file, 10
- check_unit, 11
- closest_xy, 11
- col2ncvar, 12
- cos_solar_zenith_angle, 12
- daygroup, 13
- debias.met (debias_met), 16
- debias.met.regression, 6, 13
- debias_met, 16
- download.Ameriflux, 17
- download.AmerifluxLBL, 17
- download.CRUNCEP, 19
- download.ERA5.old, 20
- download.FACE, 21
- download.Fluxnet2015, 22
- download.FluxnetLaThuile, 23
- download.Geostreams, 24, 59
- download.GFDL, 25
- download.GLDAS, 26
- download.ICOS, 27
- download.MACA, 28
- download.MERRA, 29
- download.MsTMIP_NARR, 30
- download.NARR, 31
- download.NARR_site, 32
- download.NEONmet, 33
- download.NLDAS, 34
- download.NOAA_GEFS, 35
- download.PaleON, 37
- download.PaleON_ENS, 37
- download.raw.met.module, 38
- download.US_WCr, 40
- download.US_Wlef, 40
- download_NOAA_GEFS_EFI, 41
- downscale_one_cfmet_day, 42
- downscale_repeat_6hr_to_half_hrly, 42
- downscale_repeat_6hr_to_hrly, 43
- downscale_ShortWave_to_half_hrly, 43
- downscale_ShortWave_to_hrly, 44
- downscale_solar_geom, 45
- downscale_solar_geom_halfhour, 45
- downscale_spline_to_half_hrly, 46
- downscale_spline_to_hrly, 46
- equation_of_time, 47
- ERA5_met_process, 47
- exner, 48

- extract.local.CMIP5, 49
- extract.local.NLDAS, 50
- extract.nc, 51
- extract.nc.ERA5, 52
- gen.subdaily.models, 53, 65, 83, 84, 90, 95, 101, 104
- generate_narr_url, 55
- get.es, 55
- get.lv, 56
- get.ncvector, 56
- get.rh, 57
- get.vpd, 58
- get_cf_variables_table, 58
- get_clowderauth, 59
- get_NARR_thredds, 59
- get_narr_url, 60
- half_hour_downscale, 61
- <https://confluence.ecmwf.int/display/CKB/ERA5+data+documentation#ERA5data+documentation-Spatialgrid,53>
- latlon2lcc, 61
- latlon2narr, 62
- lightME, 63
- list, 63
- lm_ensemble_sims, 54, 63, 83, 84, 90, 95, 101, 104
- load.cfmet, 9, 65
- merge_met_variable, 66
- met.process, 67
- met.process.stage, 68
- met2CF.ALMA, 69
- met2CF.Ameriflux, 69
- met2CF.AmerifluxLBL, 70
- met2CF.csv, 71
- met2CF.ERA5, 73
- met2CF.FACE, 74
- met2CF.Geostreams, 75
- met2CF.ICOS, 76
- met2CF.NARR, 77
- met2CF.PaleON, 78
- met2CF.PaleONregional, 78
- met_temporal_downscale.Gaussian_ensemble, 81
- metgapfill, 79
- metgapfill.NOAA_GEFS, 80
- model.train, 54, 65, 82, 84, 90, 95, 101, 104
- narr_all_vars (narr_flx_vars), 83
- narr_flx_vars, 83
- narr_sfc_vars (narr_flx_vars), 83
- nc.merge, 54, 65, 83, 83, 90, 95, 101, 104
- nc_create, 19, 26, 29, 51, 76
- ncdf4, 15, 16, 66, 82, 84, 90, 99, 105
- ncvar_def, 19, 26, 29, 51
- noaa_grid_download, 85
- noaa_stage2, 85
- par2ppfd, 86
- pecan_standard_met_table, 87
- permute.nc, 87
- post_process, 88
- predict_subdaily_met, 54, 65, 83, 84, 88, 95, 101, 104
- prepare_narr_year, 90
- process_gridded_noaa_download, 91
- read.register, 93
- read_narr_var, 93
- rh2qair, 94
- rh2rv (rh2qair), 94
- save.betas, 54, 65, 83, 84, 90, 94, 95, 101, 104
- save.model, 54, 65, 83, 84, 90, 95, 95, 101, 104
- site.lst, 96
- site_from_tag, 96
- solarMJ2ppfd, 97
- spin.met, 98
- split_wind, 99
- sprintf, 7, 8
- step_means, 100
- subdaily_pred, 54, 65, 83, 84, 90, 95, 101, 104
- sw2par, 102
- sw2ppfd, 102
- temporal.downscale.functions, 54, 65, 83, 84, 90, 95, 101, 103
- temporal_downscale, 104
- temporal_downscale_half_hour (half_hour_downscale), 61
- upscale_met, 105

`wide2long`, [105](#)

`write_noaa_gefs_netcdf`, [106](#)