Package 'jcext'

January 22, 2018

Type Package

Title Extended Classification of Weather Types

Version 0.1

Depends R (>= 3.2.3)

Encoding UTF-8

Author Noelia Otero

Maintainer Noelia Otero <noeli1680@gmail.com>

Description Provides a gridded classification of weather types by applying the Jenkinson and Collison classification. For a given region (it can be either local region or the whole map),it computes at each grid the 11 weather
types during the period considered for the analysis. See Otero et al., (2017) <doi:10.1007/s00382-017-3705-y> for more information.

License GPL (>= 2)

LazyData TRUE

Imports parallel, stringr, RColorBrewer, graphics, grDevices, maps, rworldmap, ggplot2, sp

Suggests ncdf4

RoxygenNote 6.0.1.9000

NeedsCompilation no

Repository CRAN

Date/Publication 2018-01-22 11:25:58 UTC

R topics documented:

`calculate_cwt` *calculate_cwt*

Description

Applies the rules to obtain the weather types. (more details, see Jones et al. 1997)

Usage

```
calculate_cwt(Z, TF, directionflow, G, thr = 6, Gth = 30)
```

Arguments

`Z`	Data frame with the total shear vorticity and dates.
`TF`	Data frame with resultant flow and dates.
`directionflow`	Data frame with the direction of the flow and dates.
`G`	Data frame with Gale days and dates.
`thr`	Numeric threshold used for Unclassified days.
`Gth`	Numeric threshold used for Gale days.

Details

As defined in the original scheme, the threshold to determine unclassified days is 6. Gale days are estimated by using a threshold of 30. If Gale days with a greater intensity (e.g. 40 or 50) is wanted, Gth must be changed. The airflow indices are used within the following rules to define the appropriate Lamb weather types.

- The appropriate direction is calculated on an eight-point compass allowing 45° per sector.
- If abs(Z) is less than TF, flow is essentially pure directional type.
- If |Z| is greater than 2TF, then the pattern is strongly cyclonic (Z > 0) or anticyclonic (Z < 0).
- If |Z| lies between TF and 2TF then the flow is partly (anti-) cyclonic and this corresponds to one of Lamb's synoptic/direction hybrid types, e.g. AE.
- If TF is less than 6 and |Z| is less than 6, there is light indeterminate flow corresponding to Lamb's unclassified type U.

Value

A list with two objects:

- Total_CWT with five groups of weather types: LWT_D, LWT_C, LWT_CH, LWT_U and LWT_G The main four groups contain the total of 27 weather types and the five list LWT_G refers to the gale days.
  - LWT_D: 8 directional types.
  - LWT_C: 1 anticyclonic and 1 cyclonic.
  - LWT_CH: 16 hybrid types.
  - LWT_U: 1 unclassified type.
- Logical values of daily occurrence of each type. The user might want to use these values to get composites with specific atmospheric fields (e.g. pressure, temperature).

See Also

`classification_jc`

`classification_jc` *classification_jc*

Description

Calculates the classification of the main weather types for one central point that is surrounded by 16-points (grid16). Wind-flow characteristics are computed for the daily pressure field according to the rules proposed by the original Jenkinson and Collison classification (see Jones et al. 1993, Jones et al. 2016).

Usage

```
classification_jc(mslp, grid16, centralp, loni, lati, times, gale)
```

Arguments

`mslp`	3-Dimensional multi-array ([loni,lati,time]) with mean sea level pressure in Pa.
`grid16`	Data frame obtained in the main function (extended_jc) that contains the 16 grid-points defining the scheme. First row is for longitudes, while the second row is for latitudes.
`centralp`	Numeric that refers to the central point for which the JC classification is calculated.
`loni`	Array with longitude values.
`lati`	Array with latitude values.
`times`	Array with the dates used.
`gale`	A logical for deteriming Gale days.

Value

Daily frequencies of Weather Types and airflow indices.

References

Jones, P. D., Hulme M., Briffa K. R. (1993) *A comparison of Lamb circulation types with an objective classification scheme* Int. J. Climatol. 13: 655–663.

Jones, P. D., Harpham C, Briffa K. R. (2013) *Lamb weather types derived from Reanalysis products* Int. J. Climatol. 33: 1129–1139.

See Also

calculate_cwt

Examples

```
# Load data
data(press)
mslp  <- press$msl
loni  <- press$loni
lati  <- press$lati
times <- press$dates
# Define a central point
centralp <- c(10,50)
# Get the scheme for the  central point
grid16 <- get_jcpoints(10,50)[1:16]
classification_jc(mslp, grid16, centralp, loni, lati, times, gale=FALSE)
```

---

createindex_ncdf *createindex_ncdf*

Description

Converts 3D arrays with daily values of airflow indices to NETCDF files.

Usage

```
createindex_ncdf(cwt_out, times, centralp, path = NULL)
```

Arguments

cwt_out	List with a list for each grid point that contains two objects $CWT and $indices.
times	Numeric with the dates.
centralp	Numeric with the centralp obtained from the main program.
path	Path name to create the output file.

Value

A ncdf file with the airflow indices.

See Also

create_ncdfcwtextended_jc

Examples

```
# This is a long running example
cwtGlobal <- extended_jc(press$msl, press$loni, press$lati, press$dates, gale=FALSE, num_cores=2)
# Create ncdf file (one file with all types)
createindex_ncdf(cwtGlobal, press$dates, cwtGlobal$centralp, path = NULL)
```

---

create_ncdfcwt *create_ncdfcwt*

Description

Converts 3D arrays with daily frequencies of weather types to NETCDF files.

Usage

```
create_ncdfcwt(cwt_out, times, centralp, onefile = TRUE, path = NULL)
```

Arguments

cwt_out	List with a list for each grid point that contains two objects: CWT and indices.
times	Numeric with the dates.
centralp	Numeric with the centralp obtained from the main program.
onefile	Logical. If TRUE one single output file with all WT is created, if FALSE one file per type is created.
path	Path name to create the output file.

Value

A netcdf file with all weather types or one file per type.

See Also

createindex_ncdfextended_jc

Examples

```
# This is a long running example
cwtGlobal <- extended_jc(press$msl, press$loni, press$lati, press$dates, gale=FALSE, num_cores=2)
# Create ncdf file (one file with all types)
create_ncdfcwt(cwtGlobal, press$dates, cwtGlobal$centralp, onefile = TRUE, path = NULL)
```

`extended_jc` *extended_jc*

Description

Gets daily classification of weather types at every grid-point over the map or selected area of interest according to the Jenkison and Collison scheme.

Usage

```
extended_jc(mslp, loni, lati, times, gale = FALSE, num_cores = 2)
```

Arguments

`mslp`	3-Dimensional array ([loni,lati,time]) with mean sea level pressure in Pa.
`loni`	Array with longitude values.
`lati`	Array with latitude values.
`times`	Array with the dates used.
`gale`	Logial. If TRUE, the function returns also Gale days.
`num_cores`	Number of cores (2 by default).

Value

A list with two objects:

- A list of eleven matrix of daily frequencies of weather types ("wtypes"). Each matrix is a 3D array [loni,lati,times] and it refers to each weather type (N,NE,E,SE,S,SW,W,A,C and U).
- A list of six matrix of daily frequency of airflow indices ("indices"). Each matrix is a 3D array [loni,lati,times] and it refers to each airflow index (W,S,TF,ZW,ZS,Z and D).
- A list with the central points for which the classification is applied.

References

Otero, N., Sillmann, J. & Butler, T. *Assessment of an extended version of the Jenkinson–Collison classification on CMIP5 models over Europe* Climate Dynamics. https://doi.org/10.1007/s00382-017-3705-y

See Also

classification_jc calculate_cwt

Examples

```
# Load data
data(press)
# Get coordinates
longitudes <- press$loni
latitudes  <- press$lati
times      <- press$dates

# Example when the classification is restricted to an area
# Select longitudes and latitudes within the European domain: -10W,40E, 40N,70N
ilon  <- which(longitudes>(-10)&longitudes<40)
loni  <- longitudes[ilon]
ilat  <- which(latitudes>40&latitudes<70)
lati  <- latitudes[ilat]
cwtEU <- extended_jc(press$msl[ilon,ilat,], loni, lati, times, gale=FALSE, num_cores=2)
 ## Not run:
# Not run
# This is a long running example
# Get the classification for the whole map, all longitudes and latitudes
cwtGlobal <- extended_jc(press$msl, longitudes, latitudes, times, gale=FALSE, num_cores=2)

## End(Not run)
```

---

get_jcpoints *get_jcpoints*

Description

Computes the 16-grid points that determine the JC-scheme, based on the coordinates of the central point.

Usage

```
get_jcpoints(lon, lat)
```

Arguments

lon	Longitude of the central point.
lat	Latitude of the central point.

Details

The function excludes the poles and the equatorial areas between 25S-25N.

Value

A data frame with the coordinates of the 16-grid points and the central point.

See Also

plot_jcscheme

---

plot_freqmap_wtypes *plot_freqmap_wtypes*

Description

Visualises absolute frequencies of the 11 main types from the extended_jc over the period.

Usage

```
plot_freqmap_wtypes(mat, loni, lati, all.types = TRUE, mytype = NULL,
  center = TRUE)
```

Arguments

mat	Matrix output from extended_jc [loni,lati,time]
loni	Vector with longitude values must be -180, 180
lati	Latitude values
all.types	Logical. If TRUE all weather types are plotted in the same plot
mytype	Character with the name of the weather type wanted (i.g. N,NE,E,SE,S,SW,W,A,C and U)
center	Logical. If TRUE a center map is plotted

Value

A ggplot2 map

See Also

extended_jc

Examples

```
## Not run:
library(jcext)
# This is a long running example for plotting results for all types globally
cwtGlobal <- extended_jc(press$msl, press$loni,press$lati, press$dates, gale=FALSE, num_cores=2)
wtypesGlobal <- cwtGlobal$wtypes
plot_freqmap_wtypes(wtypesGlobal,press$loni, press$lati, all.types = TRUE,mytype = NULL ,center = T)
# Plot the global results only for one type
plot_freqmap_wtypes(wtypesGlobal,press$loni, press$lati, all.types = FALSE,mytype = "C" ,center = T)
```

```
## End(Not run)
```

`plot_jcscheme` *Plot the classification scheme*

Description

Visualises the original Jenkinson & Collison scheme for one given central point surrounded by the 16-points.

Usage

```
plot_jcscheme(centralp, loni, lati, fullmap = TRUE)
```

Arguments

`centralp`	Array with the central point (lonigutde, latitude).
`loni`	Array with longitude values.
`lati`	Array with latitude values.
`fullmap`	Logical. If TRUE a fullmap is plotted, if FALSE, only the region selected is plotted.

Details

The map shows the scheme over the whole map, or either it shows the scheme over the selected region. For that, the maximum and mininum coordinates are defined as: Maximum longitude, by default defined by the points: x6, x10 or 14. Minimum longitude, by default defined by the points: x3, x7 or 11. Maximum latitude, by default defined by the points: x1 or x2. Minimum latitude, by default defined by the points: x15 or 16.

Value

A plot device

See Also

`get_jcpoints`

Examples

```
# Visualise the scheme for one point
library(jcext)
# Define a central point
mycentral <- c(10,50)
# load the data to get coordinates
data(press)
# Visualise the whole map
plot_jcscheme(mycentral,press$loni,press$lati,fullmap=TRUE)
# Visualise the region
plot_jcscheme(mycentral,press$loni,press$lati,fullmap=FALSE)
```

`press` *Mean Sea Level pressure files*

Description

Data from a ERA-Interim reanalysis data set downloaded from ECMWF (http://apps.ecmwf.int/datasets/data/interim-full-daily/levtype=sfc/). This data corresponds to global daily values of mean sea level pressure with 2.5 x 2.5 resolution for January 2000.

Usage

```
data(press)
```

Format

A list with values of pressure and coordinates (longitude, latitude, time)

msl mean sea level pressure values, "Pa"

longitude 144

latitude 73

times 366, one year (2000)

References

Dee et al. (2011) *The ERA-Interim reanalysis: configuration and performance of the data assimilation system. Q.J.R. Meteorol. Soc., 137: 553–597. doi:10.1002/qj.828*

Examples

```
data(press)
mslp  <- press$msl
loni  <- press$loni
lati  <- press$lati
times <- press$dates
```

read_ncdata *read_ncdata*

Description

Reads a ncdf input file to extract the input for the classification: pressure field, longitudes, latitudes and dates. Absolute time is required to read the dates properly.

Usage

```
read_ncdata(ncinput, nam_coor, units)
```

Arguments

ncinput	A NETCDF file with pressure field.
nam_coor	Names of space and time coordinates
units	Units required (Pa or hPa).

Value

A list with:

- A 3D-array of mean sea level (or pressure field) as [lon,lat,times]. The units returned as hPa.
- A numeric with longitudes values.
- A numeric with latitudes values.
- A numeric with dates values.

Index