Package 'eixport'

March 1, 2018

Title Export Emissions to Atmospheric Models

Version 0.3.0

Description

Emissions are mass of pollutants released into the atmosphere. Air quality models needs emission data, with spatial and temporal distribution, to represent air pollutant concentrations. 'eixport' creates inputs for the air quality models Weather Research and Forecasting model coupled to Chemistry 'WRF-Chem' Grell et al (2005) <doi:10.1016/j.atmosenv.2005.04.027>, Brazilian Regional Atmospheric Modeling System with a simple photochemical module 'BRAMS-SPM' Freitas et al (2005) <doi:10.1016/j.atmosenv.2005.07.017> and a line source dispersion model 'RLINE' Snyder et al (2013) <doi:10.1016/j.atmosenv.2013.05.074>.

License MIT + file LICENSE

URL https://github.com/atmoschem/eixport

BugReports https://github.com/atmoschem/eixport/issues/

Depends R (>= 2.10)

Imports sf, ncdf4, raster, sp

Encoding UTF-8

LazyData no

RoxygenNote 6.0.1

Suggests testthat, covr

NeedsCompilation no

Author Sergio Ibarra-Espinosa [aut, cre]
(<https://orcid.org/0000-0002-3162-1905>),
Daniel Schuch [aut] (<https://orcid.org/0000-0001-5977-4519>)

Maintainer Sergio Ibarra-Espinosa <sergio.ibarra@usp.br>

Repository CRAN

Date/Publication 2018-03-01 09:53:06 UTC

R topics documented:

`eixport` *eixport: a package for exporting emissions*

Description

This package provides functions to export emissions to other models

Details

1) Characteristics

This package includes functions torelated to WRF-Chem, Brams and RLINE. Also, includes the data:

1. data("emis_opt")
2. data("emisco")
3. data("Lights")
4. data("rawprofile")

`emisco` *Emissions from VEIN demo*

Description

Emissions iwth units for R-LINE

Usage

```
data(emisco)
```

Format

A sf object of lines with 288 rows and 15 variables:

ldv Light Duty Vehicles (1/h)

hdv Heavy Duty Vehicles (1/h)

lkm Length of the link (km)

ps Peak Speed (km/h)

ffs Free Flow Speed (km/h)

tstreet Type of street

lanes Number of lanes per link

capacity Capacity of vehicles in each link (1/h)

tmin Time for travelling each link (min)

V8 Emissions

xmin Initial x coordinates

xmax Ending x coordinates

ymin Initial y coordinates

ymax Ending y coordinates

geometry geometry column of the sf object data(emisco)

Source

https://github.com/ibarraespinosa/vein

emis_opt *List of WRF emission species*

Description

Emission package definitions from WRF 3.9.1.1

Usage

```
data(emis_opt)
```

Format

A list of emision variables names, same number than emis_opt in namelist.

Note

look to the number of aerosol of the emis_opt in wrf domumentation / code.

Author(s)

Daniel Schuch

Source

http://www2.mmm.ucar.edu/wrf/users/download/get_source.html

See Also

to_wrf

Examples

```
## Not run:
# Do not run
dir.create("EMISS")
data(emis_opt)
names(emis_opt)
wrf_create(wrfinput_dir = system.file("extdata", package = "eixport"),
           wrfchemi_dir = "EMISS",
           variaveis    = emis_opt["eradmsorg"],
           n_aero       = 17)

## End(Not run)
```

`Lights` *Spatial distribution exemple*

Description

Spatial distribution for veicular emissions based on an image of persistent lights of the Defense Meteorological Satellite Program (DMSP) for 5 Brazilian states (Sao Paulo, Rio de Janeiro, Mato Grosso, Santa Catarina e Parana).

Usage

```
data(Lights)
```

Format

A matrix of spatial distribution

Details

https://en.wikipedia.org/wiki/Defense_Meteorological_Satellite_Program

Author(s)

Daniel Schuch

Source

https://ngdc.noaa.gov/eog/dmsp/downloadV4composites.html

See Also

to_wrf

Examples

```
## Not run:

dir.create(file.path(tempdir(), "EMISS"))
wrf_create(wrfinput_dir = system.file("extdata", package = "eixport"),
          wrfchemi_dir = file.path(tempdir(), "EMISS"),
          frames_per_auxinput5 = 24)

# get the name of created file
files <- list.files(path = file.path(tempdir(), "EMISS"),
                    pattern = "wrfchemi",
                    full.names = TRUE)

data(Lights)
```

```
perfil <- c(0.010760058, 0.005280596, 0.002883553, 0.002666932,
           0.005781312, 0.018412838, 0.051900411, 0.077834636,
           0.067919758, 0.060831614, 0.055852868, 0.052468599,
           0.050938043, 0.051921718, 0.052756244, 0.052820165,
           0.058388406, 0.072855890, 0.075267137, 0.063246412,
           0.042713523, 0.029108975, 0.022091855, 0.015298458)

plot(perfil, ty = "l", col= "purple", xlab = "Hour", main = "Time profile",
    ylab = "Weight", axes = FALSE, xlim = c(0, 24))
axis(2)
axis(1, at = c(0, 6, 12, 18, 24),
    labels = c("00:00","06:00","12:00","18:00","00:00"))

to_wrf(Lights, files[1], total = 1521983, profile = perfil, names = "E_CO")

## End(Not run)
```

`rawprofile` *Raw profile*

Description

Raw profile

Usage

```
data(rawprofile)
```

Format

A matrix with 1 column and 168 rows
data(rawprofile)

`to_brams_spm` *Inputs for BRAMS SPM*

Description

Create inputs for SPM brams. The inputs consists in data-frame or a lists of data-frames with daily emissions (mol/day), lat, long. Also, including a functions describing the hourly profile.

Usage

```
to_brams_spm(sdf, epsg = 4326)
```

Arguments

sdf	Grid emissions, which can be a SpatialPolygonsDataFrame or polygon grid class sf' including the hourly emissions in mol/h for 24 hours. The object can also be a list of objects SpatialPolygonsDataFrame or Spatial Features polygon grid class 'sf'.
epsg	Coordinate reference system, e.g: "4326". Used to transform the coordinates of the output.

Value

data-frame of daily gridded emissions, lat, long and a message with function.

Note

When the input os class 'Spatial', they are converted to 'sf'. If the input is a data-frame, the output is a data-frame. If he input is a list, the output is a list.

Author(s)

Sergio Ibarra

References

SPM BRAMS: FREITAS, E. MARTINS, L., SILVA, P. and ANDRADE, M. A simple photochemical module implemented in rams for tropospheric ozone concentration forecast in the metropolitan area of são paulo, brazil: Coupling and validation. Atmospheric Environment, Elsevier, n. 39, p. 6352–6361, 2005.

Examples

```
## Not run:
# Do not run


## End(Not run)
```

---

to_rline *Export emissions to other formats*

Description

Export Emissions object according format of file 'Sources.txt' of the model R-LINE

Usage

```
to_rline(X_b, Y_b, Z_b, X_e, Y_e, Z_e, dCL, sigmaz0, lanes, Emis, Hw1, dw1, Hw2,
  dw2, Depth, Wtop, Wbottom, experimental = FALSE)
```

Arguments

X_b	initial x coordinates.
Y_b	initial y coordinates.
Z_b	initial meters above sea level (m).
X_e	final x coordinates.
Y_e	final y coordinates.
Z_e	final meters above sea level (m).
dCL	offset distance for each source relative to the centerline.
sigmaz0	vertical dispersion (m).
lanes	number of lanes at each street.
Emis	Column with the emissions whose unit must be g/ms.
Hw1	Height of the barrier 1 (m).
dw1	Distance to barrier 1 (m).
Hw2	height of the barrier 2 (m).
dw2	Distance to barrier 2 (m).
Depth	Depth of the depression. USed for depressed roadway (m).
Wtop	width of the opening at the top of the depression (m).
Wbottom	width of the roadway at the bottom of the depression (m).
experimental	Boolean argument to denotes the use of the experimental features (TRUE) or not (FALSE).

Value

Data-frame file with format for R-LINE model.

Note

Michelle G. Snyder, Akula Venkatram, David K. Heist, Steven G. Perry, William B. Petersen, Vlad Isakov, RLINE: A line source dispersion model for near-surface releases, In Atmospheric Environment, Volume 77, 2013, Pages 748-756, ISSN 1352-2310, https://doi.org/10.1016/j.atmosenv.2013.05.074.

Examples

```
## Not run:
# Do not run
data(emisco)
Source <- to_rline(X_b = emisco$xmin,
               Y_b = emisco$ymin,
               Z_b =0,
               X_e = emisco$xmin,
               Y_e = emisco$ymin,
               Z_e =0,
               dCL = 0,
               Emis = emisco$V8,
```

```
                sigmaz0 = 2,
                lanes = emisco$lanes)
head(Source)
write.table(x = Source, file = paste0(tempdir(), "/Sources.txt"),
row.names = FALSE, sep = " ", quote = FALSE)

## End(Not run)
```

`to_wrf` *Combine total/spasial/temporal/split and write emission to file*

Description

Function to expand, split and write emissions. The input is expand into time by profile and split between variables with diferent weights.

Usage

```
to_wrf(x, file = file.choose(), total = NA, norm = F, profile = 1,
  names = NA, weights = 1, verbose = T)
```

Arguments

`x`	matrix or array of emissions of spacial weights
`file`	emission file name
`total`	total of emited specie
`norm`	if the spacial weights need to be normalized
`profile`	temporal profile to expand the emissions
`names`	species to be write
`weights`	weight of eath specie
`verbose`	display adicional information

Note

length(profile) must be the number of times in the emission file (value of frames_per_auxinput5 if wrf_create() was used to create this file).

total is a additional way to calculate or correct the total emissions

sum(profile) = 1 and sum(weights) = 1 to conserve mass

names and weights must have the same length

Author(s)

Daniel Schuch

See Also

wrf_create, wrf_get,wrf_profile and wrf_plot

Examples

```
## Not run:
dir.create(file.path(tempdir(), "EMISS"))
wrf_create(wrfinput_dir = system.file("extdata", package = "eixport"),
          wrfchemi_dir = file.path(tempdir(), "EMISS"),
          frames_per_auxinput5 = 24)

# get the name of created file
files <- list.files(path = file.path(tempdir(), "EMISS"),
                    pattern = "wrfchemi",
                    full.names = TRUE)

data(Lights)

perfil <- c(0.010760058, 0.005280596, 0.002883553, 0.002666932,
            0.005781312, 0.018412838, 0.051900411, 0.077834636,
            0.067919758, 0.060831614, 0.055852868, 0.052468599,
            0.050938043, 0.051921718, 0.052756244, 0.052820165,
            0.058388406, 0.072855890, 0.075267137, 0.063246412,
            0.042713523, 0.029108975, 0.022091855, 0.015298458)

plot(perfil, ty = "l", col= "purple", xlab = "Hour", main = "Time profile",
    ylab = "Weight", axes = FALSE, xlim = c(0, 24))
axis(2)
axis(1, at = c(0, 6, 12, 18, 24),
    labels = c("00:00","06:00","12:00","18:00","00:00"))

to_wrf(Lights, files[1], total = 1521983, profile = perfil, names = "E_CO")

## End(Not run)
```

---

wrf_create *Create emission files to the WRF-Chem*

Description

Create an emission file from wrfinput

Usage

```
wrf_create(wrfinput_dir = "", wrfchemi_dir = "", domains = 1,
  frames_per_auxinput5 = 1, auxinput5_interval_m = 60, day_offset = 0,
  io_style_emissions = 2, kemit = 1, variaveis = c("E_NO2", "E_NO",
  "E_TOL", "E_XYL", "E_ALD", "E_ALDX", "E_SO2", "E_CO", "E_OLT", "E_OLI",
  "E_OL2", "E_NH3", "E_ISO", "E_HCL", "E_HCHO", "E_ETH", "E_CH3OH", "E_C2H5OH",
```

```
"E_HC3", "E_HC5", "E_HC8", "E_KET", "E_ORA2", "E_CSL", "E_TERP", "E_PM25I",
"E_PM25J", "E_ECI", "E_ECJ", "E_ORGI", "E_ORGJ", "E_SO4I", "E_SO4J", "E_NO3J",
"E_NO3I", "E_SO4C", "E_NO3C", "E_ORGC", "E_ECC", "E_PM10"), n_aero = 15,
COMPRESS = NA, force_ncdf4 = FALSE, verbose = FALSE)
```

Arguments

wrfinput_dir	folder with the wrfinput file(s)
wrfchemi_dir	output folder
domains	domain or domains to process
frames_per_auxinput5	value from wrf &time_control namelist.input, number of times in a single emission file
auxinput5_interval_m	value from wrf &time_control namelist.input, interval in minutes between different times
day_offset	number of days (can be a fracion) to create multiple files
io_style_emissions	from wrf &chem namelist.input
kemit	from wrf &chem namelist.input, number of levels of the emission file
variaveis	emission species, can be used data(emis_opt)
n_aero	number of aerosol species
COMPRESS	integer between 1 (least compr) and 9 (most compr) or NA for no compression
force_ncdf4	force NetCDF4 format
verbose	print file info

Note

to use io_style_emissions = 1, use day_offset increased by 0.5 (to increase 12h)

Windowns users need to rename the emission files from 'wrfchemi_d01_2011-08-01_00_00_00' to 'wrfchemi_d01_2011-08-01_00:00:00' before run wrf.exe with these files

Author(s)

Daniel Schuch

See Also

to_wrf and emis_opt

Examples

```
## Not run:
# Do not run

# emissions for a 1 day forecast for domains 1 and 2
```

```
dir.create(file.path(tempdir(), "EMISS"))

wrf_create(wrfinput_dir         = system.file("extdata", package = "eixport"),
           wrfchemi_dir         = file.path(tempdir(), "EMISS"),
           domains              = 1:2,
           frames_per_auxinput5 = 24,
           auxinput5_interval_m = 60,
           day_offset           = 0,
           verbose              = TRUE)

# emission for the last timestep

wrf_create(wrfinput_dir         = system.file("extdata", package = "eixport"),
           wrfchemi_dir         = file.path(tempdir(), "EMISS"),
           domains              = 1:2,
           frames_per_auxinput5 = 1,
           auxinput5_interval_m = 60,
           day_offset           = 1,
           verbose              = TRUE)

## End(Not run)
```

`wrf_get` *Function to read variables of emission files*

Description

Read a variable

Usage

```
wrf_get(file = file.choose(), name = NA, as_raster = F,
  raster_crs = "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs")
```

Arguments

`file`	name of file interactively (default) or specified
`name`	name of the variable (any variable)
`as_raster`	return a raster instead of a array
`raster_crs`	crs to used if as_raster is TRUE

Format

array or raster object

Author(s)

Daniel Schuch

See Also

wrf_plot and wrf_put

Examples

```
{

# create the folder and emission file
dir.create(file.path(tempdir(), "EMISS"))
wrf_create(wrfinput_dir = system.file("extdata", package = "eixport"),
         wrfchemi_dir = file.path(tempdir(), "EMISS"))

# get the name of created file
files <- list.files(path = file.path(tempdir(), "EMISS"),
                    pattern = "wrfchemi",
                    full.names = TRUE)

# open, put some numbers and write
CO <- wrf_get(file = files[1], name = "E_CO")
CO[] = rnorm(length(CO))
wrf_put(file = files[1], name = "E_CO", POL = CO)
}
```

---

wrf_grid *Creates grid from wrf file*

Description

Return a Spatial Feature multipolygon or matrix

Usage

```
wrf_grid(filewrf, type = "wrfinput", matrix = F, epsg = 4326)
```

Arguments

filewrf	wrf file
type	Type or wrf file: "wrfinput" or "geo". When type is "geo", lat long comes from mass grid, XLONG_M and XLAT_M
matrix	if the output is matrix or polygon (sf)
epsg	epsg code number (see http://spatialreference.org/ref/epsg/)

Examples

```
{
# Do not run
wrf <- paste(system.file("extdata", package = "eixport"),
                          "/wrfinput_d02", sep="")
gwrf  <- wrf_grid(wrf)
plot(gwrf, axes = TRUE)
}
```

`wrf_plot` *Simple plot from wrf emission file*

Description

Create a quick plot from wrf emission file

Usage

```
wrf_plot(file = file.choose(), name = NA, time = 1, nivel = 1,
  barra = T, lbarra = 0.2, verbose = T, ...)
```

Arguments

`file`	emission file name
`name`	pollutant name
`time`	time from emission file
`nivel`	level from the emission file
`barra`	barblot if TRUE
`lbarra`	length of barplot
`verbose`	if TRUE print some information
`...`	Arguments to be passed to plot methods

Note

If the file contains levels (kemit>1), and one frame (auxinput5_interval_m = 1) time with control the level which will be ploted

In case of a error related to plot.new() margins lbarra must be adjusted

Author(s)

Daniel Schuch

See Also

`wrf_get` and `wrf_create`

Examples

```
{

dir.create(file.path(tempdir(), "EMISS"))
wrf_create(wrfinput_dir = system.file("extdata", package = "eixport"),
          wrfchemi_dir = file.path(tempdir(), "EMISS"))

# get the name of created file
files <- list.files(path = file.path(tempdir(), "EMISS"),
                    pattern = "wrfchemi",
                    full.names = TRUE)

# open, put some numbers and write
wrf_plot(files[1], "E_CO")
}
```

`wrf_profile` *Create a spatial profile from a wrf emission file and a dataframe with*

Description

returns a traffic intensity profile (based on wrf file Times) and a traffic intensity data.frame

Usage

```
wrf_profile(x, file, verbose = T)
```

Arguments

`x`	data.frame of intenticy of traffic by hours (rows) and weekdays (columns)
`file`	emission file name
`verbose`	display adicional information

Format

a numeric vector

Author(s)

Daniel Schuch

See Also

`wrf_create` and `to_wrf`

Examples

```
## Not run:
# Do not run

# Profile based on Sao Paulo tunnel experiments
data(rawprofile)
rawprofile  <- matrix(rawprofile, nrow = 24, byrow = TRUE)
rawprofile <- as.data.frame(rawprofile)
names(rawprofile) <- c("Sunday","Monday","Tuesday","Wednesday","Thursday",
                       "Friday","Saturday")
row.names(rawprofile) <- c("00:00","01:00","02:00","03:00","04:00","05:00",
                           "06:00","07:00","08:00","09:00","10:00","11:00",
                           "12:00","13:00","14:00","15:00","16:00","17:00",
                           "18:00","19:00","20:00","21:00","22:00","23:00")

print(rawprofile)

# create the folder and emission file
dir.create(file.path(tempdir(), "EMISS"))
wrf_create(wrfinput_dir = system.file("extdata", package = "eixport"),
          wrfchemi_dir = file.path(tempdir(), "EMISS"),
          frames_per_auxinput5 = 24)

files <- list.files(path = file.path(tempdir(), "EMISS"),
                    pattern = "wrfchemi",
                    full.names = TRUE)

profile <- wrf_profile(rawprofile,files[1])

plot(profile, ty="l", lty = 2, axe = FALSE,
     main = "Traffic Intensity for Sao Paulo", xlab = "hour")
axis(2)
axis(1, at = 0.5 + c(0, 6, 12, 18, 24),
    labels = c("00:00","06:00","12:00","18:00", "00:00"))

## End(Not run)
```

`wrf_put` *Function to write variables in emission files*

Description

Extract variable

Usage

```
wrf_put(file = file.choose(), name = NA, POL)
```

Arguments

file	name of file interactively (default) or specified
name	name of the variable (any variable)
POL	input

Author(s)

Daniel Schuch

See Also

wrf_plot and wrf_get

Examples

```
{
# create the folder and emission file
dir.create(file.path(tempdir(), "EMISS"))
wrf_create(wrfinput_dir = system.file("extdata", package = "eixport"),
          wrfchemi_dir = file.path(tempdir(), "EMISS"))

# get the name of created file
files <- list.files(path = file.path(tempdir(), "EMISS"),
                    pattern = "wrfchemi",
                    full.names = TRUE)

# open, put some numbers and write
CO <- wrf_get(file = files[1], name = "E_CO")
CO[] = rnorm(length(CO))
wrf_put(file = files[1], name = "E_CO", POL = CO)
}
```

Index