Package ‘EmissV’

November 16, 2025

Title Tools for Create Emissions for Air Quality Models

Date 2025-11-14

Version 0.666.0.0

Maintainer Daniel Schuch <underschuch@gmail.com>

Description Processing tools to create emissions for use in numerical air quality models. Emissions can be calculated both using emission factors and activity data (Schuch et al 2018) <doi:10.21105/joss.00662> or using pollutant inventories (Schuch et al., 2018) <doi:10.30564/jasr.v1i1.347>. Functions to process individual point emissions, line emissions and area emissions of pollutants are available as well as methods to incorporate alternative data for Spatial distribution of emissions such as satellite images (Gavidia-Calderon et. al, 2018) <doi:10.1016/j.atmosenv.2018.09.026> or openstreetmap data (Andrade et al, 2015) <doi:10.3389/fenvs.2015.00009>.

Depends R (>= 3.4)

Imports ncdf4, units(>= 0.5-1), raster, sf, methods, data.table

Suggests testthat (>= 2.1.0), covr, lwgeom

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

URL https://atmoschem.github.io/EmissV/

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

NeedsCompilation no

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

Repository CRAN

Date/Publication 2025-11-16 20:30:02 UTC

Contents

`areaSource` *Distribution of emissions by area*

Description

Calculate the spatial distribution by a raster masked by shape/model grid information.

Usage

```
areaSource(s, r, grid = NA, name = "", as_frac = FALSE, verbose = TRUE)
```

Arguments

`s`	input shape object
`r`	input raster object
`grid`	grid with the output format
`name`	area name
`as_frac`	return a fraction instead of the raster value
`verbose`	display additional data

Value

a raster object containing the spatial distribution of emissions

Source

Example data from Defense Meteorological Satellite Program (DMSP)

Examples

```
shape  <- raster::shapefile(paste(system.file("extdata", package = "EmissV"),
                            "/BR.shp",sep=""),verbose = FALSE)
shape  <- shape[22,1] # subset for Sao Paulo - BR
raster <- raster::raster(paste(system.file("extdata", package = "EmissV"),
                         "/dmsp.tiff",sep=""))
grid  <- gridInfo(paste(system.file("extdata", package = "EmissV"),"/wrfinput_d02",sep=""))
SP     <- areaSource(shape,raster,grid,name = "SPMA")

raster::plot(SP,ylab="Lat",xlab="Lon",
             main ="Spatial Distribution by Lights for Sao Paulo - Brazil")
```

---

emission *Emissions in the format for atmospheric models*

Description

Combine area sources and total emissions to model output

Usage

```
emission(
  inventory = NULL,
  grid,
  mm = 1,
  aerosol = FALSE,
  check = TRUE,
  total,
  pol,
  area,
  plot = FALSE,
  verbose = TRUE
)
```

Arguments

`inventory`	a inventory raster from read
`grid`	grid information
`mm`	pollutant molar mass
`aerosol`	TRUE for aerosols and FALSE (defoult) for gazes
`check`	TRUE (defoult) to check negative and NA values and replace it for zero
`total`	list of total emission
`pol`	pollutant name
`area`	list of area sources or matrix with a spatial distribution
`plot`	TRUE for plot the final emissions
`verbose`	display additional information

Format

matrix of emission

Value

a vector of emissions in MOL / mk2 h for gases and ug / m2 s for aerosols.

Note

if Inventory is provided, the firsts tree arguments are not be used by the function.

Is a good practice use the set_units(fe,your_unity), where fe is your emission factory and your_unity is usually g/km on your emission factory

the list of area must be in the same order as defined in vehicles and total emission.

just WRF-Chem is suported by now

See Also

totalEmission and areaSource

Examples

```
fleet  <- vehicles(example = TRUE)

EmissionFactors <- emissionFactor(example = TRUE)

TOTAL  <- totalEmission(fleet,EmissionFactors,pol = c("CO"),verbose = TRUE)

grid   <- gridInfo(paste0(system.file("extdata", package = "EmissV"),"/wrfinput_d01"))
shape  <- raster::shapefile(paste0(system.file("extdata", package = "EmissV"),"/BR.shp"))
raster <- raster::raster(paste0(system.file("extdata", package = "EmissV"),"/dmsp.tiff"))

SP     <- areaSource(shape[22,1],raster,grid,name = "SP")

e_CO   <- emission(total = TOTAL,
                   pol = "CO",
                   area = list(SP = SP),
                   grid = grid,
                   mm = 28)
```

---

emissionFactor *Tool to set-up emission factors*

Description

Return a data frame for emission for multiple pollutants.

Usage

```
emissionFactor(
  ef,
  poluttant = names(ef),
  vnames = NA,
  unit = "g/km",
  example = FALSE,
  verbose = TRUE
)
```

Arguments

`ef`	list with emission factors
`poluttant`	poluttant names
`vnames`	name of each vehicle categoy (optional)
`unit`	tring with unit from unit package, for default is "g/km"
`example`	TRUE to diaplay a simple example
`verbose`	display additional information

Value

a emission factor data frame

a emission factor data.frame for totalEmission function

See Also

`areaSource` and `totalEmission`

Examples

```
EF <- emissionFactor(example = TRUE)

# or the code for the same result
EF <- emissionFactor(ef = list(CO = c(1.75,10.04,0.39,0.45,0.77,1.48,1.61,0.75),
                               PM = c(0.0013,0.0,0.0010,0.0612,0.1052,0.1693,0.0,0.0)),
                  vnames = c("Light Duty Vehicles Gasohol","Light Duty Vehicles Ethanol",
                        "Light Duty Vehicles Flex","Diesel Trucks","Diesel Urban Busses",
                               "Diesel Intercity Busses","Gasohol Motorcycles",
                               "Flex Motorcycles"))
```

gridInfo *Read grid information from a NetCDF file*

Description

Return a list containing information of a regular grid / domain

Usage

```
gridInfo(
  file = file.choose(),
  z = FALSE,
  missing_time = "1984-03-10",
  verbose = TRUE
)
```

Arguments

file	file name/path to a wrfinput, wrfchemi or geog_em file
z	TRUE for read wrfinput vertical coordinades
missing_time	time if the variable Times is missing
verbose	display additional information

Value

a list with grid information from air quality model

Note

just WRF-Chem is suported by now

Examples

```
grid_d1 <- gridInfo(paste(system.file("extdata", package = "EmissV"),
                                  "/wrfinput_d01",sep=""))
grid_d2 <- gridInfo(paste(system.file("extdata", package = "EmissV"),
                                  "/wrfinput_d02",sep=""))
grid_d3 <- gridInfo(paste(system.file("extdata", package = "EmissV"),
                                  "/wrfinput_d03",sep=""))
names(grid_d1)
# for plot the shapes
shape   <- raster::shapefile(paste0(system.file("extdata", package = "EmissV"),
                                         "/BR.shp"))
raster::plot(shape,xlim = c(-55,-40),ylim = c(-30,-15), main="3 nested domains")
axis(1); axis(2); box(); grid()
lines(grid_d1$boundary, col = "red")
text(grid_d1$xlim[2],grid_d1$Ylim[1],"d1",pos=4, offset = 0.5)
lines(grid_d2$boundary, col = "red")
```

```
text(grid_d2$xlim[2],grid_d2$Ylim[1],"d2",pos=4, offset = 0.5)
lines(grid_d3$boundary, col = "red")
text(grid_d3$xlim[1],grid_d3$Ylim[2],"d3",pos=2, offset = 0.0)
```

---

hourly *Temporal hourly emission profiles by sector for Brazil*

Description

Set of hourly profiles that represents the mean activity for each hour (local time) of the week (currently the profile have the same emissions for different week days).

ENE Energy sector

IND_FUEL Industry and Fuel production sectors

RES_COM Residential and Comercial sectors

AGR Agriculture sector

REF Oil refineryes and ships (constant)

AW Waste Burn

TRO_PC Passanger cars

TRO_HGV Heavy Duty vehicles

Usage

```
data(hourly)
```

Format

A list of data frames with activity by hour and weekday.

Details

profiles from Schuch et al. (2026A) based on profiles from Europe and modifications considering local data from Brazil.

Note

The profile is normalized by days (but is balanced for a complete week) it means diary_emission x profile = hourly_emission.

References

Daniel Schuch, Y. Zhang, S. Ibarra-Espinosa, M. F. Andradede, M. Gavidia-Calderónd, and M. L. Belle. Multi-Year Evaluation and Application of the WRF-Chem Model for Two Major Urban Areas in Brazil - Part I: Initial Application and Model Improvement. Atmospheric Environment, 2026A. doi:10.1016/j.atmosenv.2025.121577

Examples

```
# load the data
data(hourly)
# plot the data
colors <- c("#A60026","#EF603D","#d1b64b","#8eb63b",
            "#56B65F","#AAAAAA","#7bc2f0","#992299")

plot(hourly$ENE$Sun,ty = "l", ylim = c(0,3),axe = FALSE,xlab='',
     ylab='',col = colors[1], lwd = 2,
     main = "Hourly emission profile by sector")
points(hourly$ENE$Sun,col = colors[1], pch = 20, cex = 2)
lines(hourly$IND_FUEL$Sun, col = colors[2], lwd = 2)
points(hourly$IND_FUEL$Sun,col = colors[2], pch = 20, cex = 2)
lines(hourly$RES_COM$Sun, col = colors[3], lwd = 2)
points(hourly$RES_COM$Sun,col = colors[3], pch = 20, cex = 2)
lines(hourly$AGR$Sun, col = colors[4], lwd = 2)
points(hourly$AGR$Sun,col = colors[4], pch = 20, cex = 2)
lines(hourly$REF$Sun, col = colors[5], lwd = 2)
points(hourly$REF$Sun,col = colors[5], pch = 20, cex = 2)
lines(hourly$WB$Sun, col = colors[6], lty = 1, lwd = 2)
lines(hourly$TRO_PC$Sun, col = colors[7], lwd = 2)
points(hourly$TRO_PC$Sun,col = colors[7], pch = 20, cex = 2)
lines(hourly$TRO_HGV$Sun, col = colors[8], lty = 1, lwd = 2)
points(hourly$TRO_HGV$Sun,col = colors[8], pch = 20, cex = 2)

axis(1,at=0.5+c(0,6,12,18,24),
     labels = c("00:00","06:00","12:00","18:00","00:00"))
axis(2,at=c(0,0.5,1.0,1.5,2.0, 2.5, 3.0))
legend('topleft',legend = c('energy production',
                            'industrial / fuel exploitation',
                            'food and paper production / residential',
                            'agriculture',
                            'oil refineries / ship',
                            'waste burn',
                            'TRO non high-emitters',
                            'TRO high-emitters'),
    bty = 'n', lty = c(1,1,1,1,1,1,1,1),
    col = colors, lwd = 2, cex = 1.2)
mtext('Intensity',2,cex = 1.5, line = 2.6)
mtext('hour (Local)',1,cex = 1.5, line = 2.6)
```

`lineSource` *Distribution of emissions by lines*

Description

Create a emission distribution from 'sp' or 'sf' spatial lines data.frame or spatial lines.

There 3 modes available to create the emission grid: - using gridInfo function output (defoult) - using the patch to "wrfinput" (output from real.exe) file or "geo" for (output from geog.exe) - "sf" (and "sp") uses a grid in SpatialPolygons format

The variable is the column of the data.frame with contains the variable to be used as emissions, by defoult the idstribution taken into acount the lench distribution of lines into each grid cell and the output is normalized.

Usage

```
lineSource(
  s,
  grid,
  as_raster = FALSE,
  type = "info",
  gcol = 100,
  grow = 100,
  variable = "length",
  verbose = TRUE
)
```

Arguments

s	SpatialLinesDataFrame of SpatialLines object
grid	grid object with the grid information or filename
as_raster	output format, TRUE for raster, FALSE for matrix
type	"info" (default), "wrfinput", "geo", "sp" or "sf" for grid type
gcol	grid points for a "sp" or "sf" type
grow	grid points for a "sp" or "sf" type
variable	variable to use, default is line length
verbose	display additional information

Value

a raster object containing the spatial distribution of emissions

Source

OpenstreetMap data avaliable https://www.openstreetmap.org/ and https://download.geofabrik.de/

See Also

gridInfo and rasterSource

Examples

```
# loading a shapefile with osm data for sao paulo metropolitan area
roads <- sf::st_read(paste0(system.file("extdata",package="EmissV"),"/streets.shp"))
d3    <- gridInfo(paste0(system.file("extdata", package = "EmissV"),"/wrfinput_d03"))

# calculating only for 2 streets
```

```
roadLength <- lineSource(roads[1:2,],d3,as_raster=TRUE)

# to generate a quick plot
raster::plot(roadLength,ylab="Lat", xlab="Lon",main="Length of roads")
# lines(road_lines)
```

`monthly` *Temporal monthly emission profile for total emissions for Brazil*

Description

Set of monthly profiles that represents the mean activity for each month of the year.

month month (1 to 12)

VOC profile for total emissions of VOC

NOx profile for total emissions of NOx

PM profile for total emissions of PM

Usage

```
data(monthly)
```

Format

A data frames with activity by month of the year.

Details

Profiles from Schuch et al. (2026B) based on WRF-Chem numerical experiments for 2012-2016 and observations available for for MASP and MARJ.

Note

The profile is normalized by month (but is balanced for a complete year) it means anual_emission x profile = monthly_emission.

References

Daniel Schuch, Y. Zhang, S. Ibarra-Espinosa, M. F. Andradede, M. Gavidia-Calderónd, and M. L. Belle. Multi-Year Evaluation and Application of the WRF-Chem Model for Two Major Urban Areas in Brazil part II: Multi-Year evaluation and urban-centric analysis. Atmospheric Environment, 2026B. doi:10.1016/j.atmosenv.2025.121632

Examples

```
# load the data
data(monthly)
# make a plot
cols <- c("NOx" = "#28B2E0","VOC" = "#66E024","PM" = "#960606")

plot(NA, xlim = c(1,12), ylim = c(-75,130),
     xaxt = "n", xlab = "Month",main = "Profile for total emissions",
     ylab = "Monthly adjustment [%]",cex.lab = 1.4, cex.axis = 1.2)
axis(1, at = 1:12, labels = month.abb, cex.axis = 1.2)

for (v in c("NOx", "VOC", "PM")) {
  y <- monthly[[v]] * 100 - 100  # convert to % change
  points(monthly$month, y, col = cols[v], pch = 16)

  lo <- loess(y ~ month, data = monthly, span = 0.4)
  xs <- seq(1,12, length.out = 200)

  lines(xs, predict(lo, newdata = data.frame(month = xs)),
        col = cols[v], lwd = 2)
}

legend("topleft", legend = c(expression(NO[x]), "VOCs", expression(PM[2.5])),
       col = cols, pch = 16, lwd = 2, pt.cex = 1.5, bty = "n",cex = 1.2)
```

`perfil` *Temporal profile for emissions*

Description

Set of hourly profiles that represents the mean activity for each hour (local time) of the week.

LDV Light Duty vehicles

HDV Heavy Duty vehicles

PC_JUNE_2012 passenger cars counted in June 2012

PC_JUNE_2013 passenger cars counted in June 2013

PC_JUNE_2014 passenger cars counted in June 2014

LCV_JUNE_2012 light comercial vehicles counted in June 2012

LCV_JUNE_2013 light comercial vehicles counted in June 2013

LCV_JUNE_2014 light comercial vehicles counted in June 2014

MC_JUNE_2012 motorcycles counted in June 2012

MC_JUNE_2013 motorcycles counted in June 2013

MC_JUNE_2014 motorcycles counted in June 2014

HGV_JUNE_2012 Heavy good vehicles counted in June 2012

HGV_JUNE_2013 Heavy good vehicles counted in June 2013

HGV_JUNE_2014 Heavy good vehicles counted in June 2014
PC_JANUARY_2012 passenger cars counted in january 2012
PC_JANUARY_2013 passenger cars counted in january 2013
PC_JANUARY_2014 passenger cars counted in january 2014
LCV_JANUARY_2012 light comercial vehicles counted in january 2012
LCV_JANUARY_2013 light comercial vehicles counted in january 2013
LCV_JANUARY_2014 light comercial vehicles counted in january 2014
MC_JANUARY_2012 Motorcycles counted in january 2012
MC_JANUARY_2014 Motorcycles counted in january 2014
HGV_JANUARY_2012 Heavy good vehicles counted in january 2012
HGV_JANUARY_2013 Heavy good vehicles counted in january 2013
HGV_JANUARY_2014 Heavy good vehicles counted in january 2014
POW Power generation emission profile
IND Industrial emission profile
RES Residencial emission profile
TRA Transport emission profile
AGR Agriculture emission profile
SHP Emission profile for ships
SLV Solvent use emission constant profile
WBD Waste burning emisssion constant profile
PC_nov_2018 passenger cars at Janio Quadros on November 2018
HGV_nov_2018 heavy good vehicles at Janio Quadros on November 2018
TOTAL_nov_2018 total vehicle at Janio Quadros on November 2018
PC_out_2018 passenger cars at Anhanguera-Castello Branco on October 2018
MC_out_2018 Motorcycles cars at Anhanguera-Castello Branco on October 2018

Usage

```
data(perfil)
```

Format

A list of data frames with activity by hour and weekday.

Details

- Profiles 1 to 2 are from traffic count at São Paulo city from Perez Martínez et al (2014).

- Profiles 3 to 25 comes from traffic counted of toll stations located in São Paulo city, for summer and winters of 2012, 2013 and 2014.

- Profiles 26 to 33 are for different sectors from Oliver et al (2003).

- Profiles 34 to 36 are for volumetric mechanized traffic count at Janio Quadros tunnel on November 2018.

- Profiles 37 to 38 are for volumetric mechanized traffic count at Anhanguera-Castello Branco on October 2018.

Note

The profile is normalized by days (but is balanced for a complete week) it means diary_emission x profile = hourly_emission.

References

Pérez-Martínez, P. J., Miranda, R. M., Nogueira, T., Guardani, M. L., Fornaro, A., Ynoue, R., & Andrade, M. F. (2014). Emission factors of air pollutants from vehicles measured inside road tunnels in São Paulo: case study comparison. International Journal of Environmental Science and Technology, 11(8), 2155-2168.

Olivier, J., J. Peters, C. Granier, G. Pétron, J.F. Müller, and S. Wallens, Present and future surface emissions of atmospheric compounds, POET Report #2, EU project EVK2-1999-00011, 2003.

Examples

```
# load the data
data(perfil)

# function to simple view
plot.perfil <- function(per = perfil$LDV, text="", color = "#0000FFBB"){
  plot(per[,1],ty = "l", ylim = range(per),axe = FALSE,
       xlab = "hour",ylab = "Intensity",main = text,col=color)
  for(i in 2:7){
    lines(per[,i],col = color)
  }
  for(i in 1:7){
    points(per[,i],col = "black", pch = 20)
  }
  axis(1,at=0.5+c(0,6,12,18,24),labels = c("00:00","06:00","12:00","18:00","00:00"))
  axis(2)
  box()
}

# view all profiles in perfil data
for(i in 1:length(names(perfil))){
  cat(paste("profile",i,names(perfil)[i],"\n"))
  plot.perfil(perfil[[i]],names(perfil)[i])
}
```

`plumeRise` *Calculate plume rise height.*

Description

Calculate the maximum height of rise based on Brigs (1975), the height is calculated using different formulations depending on stability and wind conditions.

Usage

```
plumeRise(df, imax = 10, ermax = 1/100, Hmax = TRUE, verbose = TRUE)
```

Arguments

`df`	data.frame with micrometeorological and emission data
`imax`	maximum number of iteractions
`ermax`	maximum error
`Hmax`	use weil limit for plume rise, see details
`verbose`	display additional information

Format

data.frame with the input, rise (m) and effective higt (m)

Details

The input data.frame must contains the folloging colluns:

- z: height of the emission (m)

- r: source raius (m)

- Ve: emission velocity (m/s)

- Te: emission temperature (K)

- ws: wind speed (m/s)

- Temp: ambient temperature (K)

- h: height of the Atmospheric Boundary Layer-ABL (m)

- L: Monin-Obuhkov Lench (m)

- dtdz: lapse ration of potential temperature, used only for stable ABL (K/m)

- Ustar: atriction velocity, used only for neutral ABL (m/s)

- Wstar: scale of convectie velocity, used only for convective ABL (m/s)

Addcitionaly some combination of ws, Wstar and Ustar can produce inacurate results, Weil (1979) propose a geometric limit of 0.62 * (h - Hs) for the rise value.

Value

a data.frame with effective height of emissions for pointSource function

References

The plume rise formulas are from Brigs (1975):"Brigs, G. A. Plume rise predictions, Lectures on Air Pollution and Environmental Impact Analyses. Amer. Meteor. Soc. p. 59-111, 1975." and Arya 1999: "Arya, S.P., 1999, Air Pollution Meteorology and Dispersion, Oxford University Press, New York, 310 p."

The limits are from Weil (1979): "WEIL, J.C. Assessmet of plume rise and dispersion models using LIDAR data, PPSP-MP-24. Prepared by Environmental Center, Martin Marietta Corporation, for Maryland Department of Natural Resources. 1979."

The example is data from a chimney of the Candiota thermoelectric powerplant from Arabage et al (2006) "Arabage, M. C.; Degrazia, G. A.; Moraes O. L. Simulação euleriana da dispersão local da pluma de poluente atmosférico de Candiota-RS. Revista Brasileira de Meteorologia, v.21, n.2, p. 153-160, 2006."

Examples

```
candiota <- matrix(c(150,1,20,420,3.11,273.15 + 3.16,200,-34.86,3.11,0.33,
                     150,1,20,420,3.81,273.15 + 4.69,300,-34.83,3.81,0.40,
                     150,1,20,420,3.23,273.15 + 5.53,400,-24.43,3.23,0.48,
                     150,1,20,420,3.47,273.15 + 6.41,500,-15.15,3.48,0.52,
                     150,1,20,420,3.37,273.15 + 6.35,600, -8.85,3.37,2.30,
                     150,1,20,420,3.69,273.15 + 5.93,800,-10.08,3.69,2.80,
                     150,1,20,420,3.59,273.15 + 6.08,800, -7.23,3.49,1.57,
                     150,1,20,420,4.14,273.15 + 6.53,900,-28.12,4.14,0.97),
                     ncol = 10, byrow = TRUE)
candiota <- data.frame(candiota)
names(candiota) <- c("z","r","Ve","Te","ws","Temp","h","L","Ustar","Wstar")
row.names(candiota) <- c("08:00","09:00",paste(10:15,":00",sep=""))
candiota <- plumeRise(candiota,Hmax = TRUE)
print(candiota)
```

`pointSource` *Emissions from point sources*

Description

Transform a set of points into a grinded output

Usage

```
pointSource(emissions, grid, verbose = TRUE)
```

Arguments

`emissions`	list of points
`grid`	grid object with the grid information
`verbose`	display additional information

Value

a raster

See Also

gridInfo and rasterSource

Examples

```
d1 <- gridInfo(paste(system.file("extdata", package = "EmissV"),"/wrfinput_d01",sep=""))

p = data.frame(lat      = c(-22,-22,-23.5),
               lon      = c(-46,-48,-47  ),
               z        = c(0   ,  0,  0  ),
               emission = c(666,444,111  ) )

p_emissions <- pointSource(emissions = p, grid = d1)

raster::plot(p_emissions,ylab="Lat", xlab="Lon",
             main = "3 point sources for domain d1")
```

---

rasterSource *Distribution of emissions by a georeferenced image*

Description

Calculate the spatial distribution by a raster

Usage

```
rasterSource(r, grid, nlevels = "all", conservative = TRUE, verbose = TRUE)
```

Arguments

r	input raster object
grid	grid object with the grid information
nlevels	number of vertical levels off the emission array
conservative	TRUE (default) to conserve total mass, FALSE to conserve flux
verbose	display additional information

Value

Returns a matrix

Source

Example data is from Defense Meteorological Satellite Program (DMSP).

See Also

gridInfo and lineSource

Examples

```
grid  <- gridInfo(paste(system.file("extdata", package = "EmissV"),"/wrfinput_d01",sep=""))
x    <- raster::raster(paste(system.file("extdata", package = "EmissV"),"/dmsp.tiff",sep=""))
test  <- rasterSource(x, grid)
image(test, axe = FALSE, main = "Spatial distribution by Persistent Nocturnal Lights from DMSP")
```

---

read   *Read NetCDF data from global inventories*

Description

Read data from global inventories. Several files can be read to produce one emission output and/or can be splitted into several species

Usage

```
read(
  file = file.choose(),
  version = NA,
  coef = rep(1, length(file)),
  spec = NULL,
  year = 1,
  month = 1,
  hour = 1,
  categories,
  reproject = TRUE,
  as_raster = TRUE,
  skip_missing = FALSE,
  verbose = TRUE
)
```

Arguments

`file` file name or names (variables are summed)

`version` Character; One of of the following:

argument	**tested**	**region**	**resolution**	**projection**
EDGAR	4.32 and 5.0	Global	0.1 x 0.1 °	longlat
EDGAR_HTAPv2	2.2	Global	0.1 x 0.1 °	longlat
EDGARv8m	8.1 monthly	Global	0.1 x 0.1 °	longlat
EDGARv8	8.1	Global	0.1 x 0.1 °	longlat
GAINS	v5a	Global	0.5 x 0.5 °	longlat
RCP	RCP3PD Glb	Global	0.5 x 0.5 °	longlat

MACCITY	2010	Global	0.5 x 0.5 °	longlat
FFDAS	2.2	Global	0.1 x 0.1 °	longlat
ODIAC	2020	Global	1 x 1 °	longlat
VULCAN-y	3.0	US	1 x 1 km	lcc
VULCAN-h	3.0	US	1 x 1 km	lcc
ACES	2020	NE US	1 x 1 km	lcc
GEMS	2023	Global	0.1 x 0.1 °	longlat
GEMSm	2023	Global	0.1 x 0.1 °	longlat

`coef`	coefficients to merge different sources (file) into one emission
`spec`	numeric speciation vector to split emission into different species
`year`	scenario index (only for GAINS and VULCAN-y)
`month`	the desired month of the inventory (MACCITY, ODIAC, EDGARv8m, and GEMSm)
`hour`	hour of the emission (only for ACES and VULCAN-h)
`categories`	considered categories (for MACCITY/GAINS variable names), empty for use all
`reproject`	to project the output to "+proj=longlat" needed for emission function (only for VULCAN and ACES)
`as_raster`	return a raster (default) or matrix (with units)
`skip_missing`	return a zero emission and a warning for missing files/variables
`verbose`	display additional information

Value

Matrix or raster

Note

for EDGAR (all versions), GAINS, RCP and MACCTITY, please use flux (kg m-2 s-1) NetCDF file.

Source

Read abbout EDGAR at http://edgar.jrc.ec.europa.eu and MACCITY at http://accent.aero.jussieu.fr/MACC_metadata.php

More info for EDGARv8.1 https://edgar.jrc.ec.europa.eu/dataset_ap81 for short live species and https://edgar.jrc.ec.europa.eu/dataset_ghg80 for GHG

References

Janssens-Maenhout, G., Dentener, F., Van Aardenne, J., Monni, S., Pagliari, V., Orlandini, L., ... & Wankmüller, R. (2012). EDGAR-HTAP: a harmonized gridded air pollution emission dataset based on national inventories. European Commission Joint Research Centre Institute for Environment and Sustainability. JRC 68434 UR 25229 EUR 25229, ISBN 978-92-79-23123-0.

Lamarque, J.-F., Bond, T. C., Eyring, V., Granier, C., Heil, A., Klimont, Z., Lee, D., Liousse, C., Mieville, A., Owen, B., Schultz, M. G., Shindell, D., Smith, S. J., Stehfest, E., Van Aardenne, J., Cooper, O. R., Kainuma, M., Mahowald, N., McConnell, J. R., Naik, V., Riahi, K., and van Vuuren, D. P.: Historical (1850-2000) gridded anthropogenic and biomass burning emissions of reactive gases and aerosols: methodology and application, Atmos. Chem. Phys., 10, 7017-7039, doi:10.5194/acp-10-7017-2010, 2010.

Z Klimont, S. J. Smith and J Cofala The last decade of global anthropogenic sulfur dioxide: 2000–2011 emissions Environmental Research Letters 8, 014003, 2013

Gurney, Kevin R., Jianming Liang, Risa Patarasuk, Yang Song, Jianhua Huang, and Geoffrey Roest (2019) The Vulcan Version 3.0 High-Resolution Fossil Fuel CO2 Emissions for the United States. Nature Scientific Data.

See Also

rasterSource and gridInfo

species

Examples

```
folder <- file.path(tempdir(), "EDGARv8.1")
dir.create(folder)

url     <- "https://jeodpp.jrc.ec.europa.eu/ftp/jrc-opendata/EDGAR/datasets"
dataset <- "v81_FT2022_AP_new/NOx/TOTALS/flx_nc"
file    <- "v8.1_FT2022_AP_NOx_2022_TOTALS_flx_nc.zip"

download.file(paste0(url,"/",dataset,"/",file), paste0(folder,"/",file))

unzip(paste0(folder,"/",file),exdir = folder)

nox  <- read(file    = dir(path=folder, pattern="flx\\.nc", full.names=TRUE),
             version = "EDGARv8",
             spec    = c(E_NO  = 0.9 ,   # 90% of NOx is NO
                         E_NO2 = 0.1 ))  # 10% of NOx is NO2
# creating a color scale
cor <- colorRampPalette(colors = c(c("#031057", "#0522FC",
                                     "#7E0AFA", "#EF0AFF",
                                     "#FFA530", "#FFF957")))
raster::plot(nox$E_NO, xlab="Latitude", ylab="Longitude",
             col = cor(12),zlim = c(-6.5e-7,1.4e-5),
             main="TOTAL NO emissions from EDGARv8.1 (in g / m2 s)")

d1  <- gridInfo(paste(system.file("extdata", package = "EmissV"),"/wrfinput_d01",sep=""))
NO  <- emission(grid = d1, inventory = nox$E_NO, pol = "NO", mm = 30.01,plot = TRUE)
```

speciation *Speciation of emissions in different compounds*

Description

Distribute the total mass of estimated emissions into model species.

Usage

```
speciation(total, spec = NULL, verbose = TRUE)
```

Arguments

total	emissions from totalEmissions
spec	numeric speciation vector of species
verbose	display additional information

Value

Return a list with the daily total emission by interest area (cityes, states, countries, etc).

See Also

species

Examples

```
veic <- vehicles(example = TRUE)
EmissionFactors <- emissionFactor(example = TRUE)
TOTAL <- totalEmission(veic,EmissionFactors,pol = "PM")
pm_iag <- c(E_PM25I = 0.0509200,
            E_PM25J = 0.1527600,
            E_ECI   = 0.1196620,
            E_ECJ   = 0.0076380,
            E_ORGI  = 0.0534660,
            E_ORGJ  = 0.2279340,
            E_SO4I  = 0.0063784,
            E_SO4J  = 0.0405216,
            E_NO3J  = 0.0024656,
            E_NO3I  = 0.0082544,
            E_PM10  = 0.3300000)
PM <- speciation(TOTAL,pm_iag)
```

`species` *Species mapping tables*

Description

Set of tables for speciation:

voc_radm2_mic Volatile organic compounds for RADM2

voc_cbmz_mic Volatile organic compounds for CBMZ

voc_moz_mic Volatile organic compounds for MOZART

voc_saprc99_mic volatile organic compounds for SAPRC99

veicularvoc_radm2_iag Vehicular volatile organic compounds for RADM2 (MOL)

veicularvoc_cbmz_iag Vehicular volatile organic compounds for CBMZ (MOL)

veicularvoc_moz_iag Vehicular volatile organic compounds for MOZART (MOL)

veicularvoc_saprc99_iag Vehicular volatile organic compounds for SAPRC99 (MOL)

pm_madesorgan_iag Particulate matter for made/sorgan

pm25_madesorgan_iag Fine particulate matter for made/sorgan

nox_iag Nox split Perez Martínez et al (2014)

nox_bcom Nox split usin Ntziachristos and Zamaras (2016)

voc_radm2_edgar432 Volatile organic compounds species from EDGAR 4.3.2 for RADM2 (MOL)

voc_moz_edgar432 Volatile organic compounds species from EDGAR 4.3.2 for MOZART (MOL)

- Volatile organic compounds species map from 1 to 4 are from Li et al (2014) taken into account several sources of pollutants.

- Volatile organic compounds from vehicular activity species map 5 to 8 is a by fuel and emission process from USP-IAG tunel experiments (Rafee et al., 2017) emited by the process of exhaust (through the exhaust pipe), liquid (carter and evaporative) and vapor (fuel transfer operations).

- Particulate matter speciesmap for made/sorgan emissions 9 and 10.

- Nox split using Perez Martínez et al (2014) data (11).

- Nox split using mean of Ntziachristos and Zamaras (2016) data (12).

- Volatile organic compounds species map 13 and 14 are the corespondence from EDGAR 4.3.2 VOC specialization to RADM2 and MOZART.

Usage

```
data(species)
```

Format

List of numeric vectors with the 'names()' of the species and the values of each species.

Details

iag-voc: After estimating all the emissions of NMHC, it was used the speciation presented in (RAFEE et al., 2017). This speciation is based on tunnel measurements in São Paulo, depends on the type of fuel (E25, E100 and B5) and provides the mass of each chemical compound as mol/g. This speciation splits the NMHC from evaporative, liquid and exhaust emissions of E25, E100 and B5, into minimum compounds required for the Carbon Bond Mechanism (CBMZ) (ZAVERI; PETERS, 1999). Atmospheric simulations using the same pollutants in Brazil have resulted in good agreement with observations (ANDRADE et al., 2015).

iag-pm: data tunnel experiments at São Paulo in Perez Martínez et al (2014)

iag-nox: common NOx split for São Paulo Metropolitan area.

bcom-nox: mean of Ntziachristos and Zamaras (2016) data.

mic: from Li et al (2014).

edgar: Edgar 4.3.2 emissions Crippa et al. (2018).

Note

The units are mass ratio (mass/mass) or MOL (MOL), this last case do not change the default 'mm' into 'emission()' function.

References

Li, M., Zhang, Q., Streets, D. G., He, K. B., Cheng, Y. F., Emmons, L. K., ... & Su, H. (2014). Mapping Asian anthropogenic emissions of non-methane volatile organic compounds to multiple chemical mechanisms. Atmos. Chem. Phys, 14(11), 5617-5638.

Huang, G., Brook, R., Crippa, M., Janssens-Maenhout, G., Schieberle, C., Dore, C., ... & Friedrich, R. (2017). Speciation of anthropogenic emissions of non-methane volatile organic compounds: a global gridded data set for 1970–2012. Atmospheric Chemistry and Physics, 17(12), 7683.

Abou Rafee, S. A., Martins, L. D., Kawashima, A. B., Almeida, D. S., Morais, M. V. B., Souza, R. V. A., Oliveira, M. B. L., Souza, R. A. F., Medeiros, A. S. S., Urbina, V., Freitas, E. D., Martin, S. T., and Martins, J. A.: Contributions of mobile, stationary and biogenic sources to air pollution in the Amazon rainforest: a numerical study with the WRF-Chem model, Atmos. Chem. Phys., 17, 7977-7995, https://doi.org/10.5194/acp-17-7977-2017, 2017.

Martins, L. D., Andrade, M. F. D., Freitas, E., Pretto, A., Gatti, L. V., Junior, O. M. A., et al. (2006). Emission factors for gas-powered vehicles traveling through road tunnels in Sao Paulo, Brazil. Environ. Sci. Technol. 40, 6722–6729. doi: 10.1021/es052441u

Pérez-Martínez, P. J., Miranda, R. M., Nogueira, T., Guardani, M. L., Fornaro, A., Ynoue, R., & Andrade, M. F. (2014). Emission factors of air pollutants from vehicles measured inside road tunnels in São Paulo: case study comparison. International Journal of Environmental Science and Technology, 11(8), 2155-2168.

ANDRADE, M. d. F. et al. Air quality forecasting system for southeastern brazil. Frontiers in Environmental Science, Frontiers, v. 3, p. 1–12, 2015.

Crippa, M., Guizzardi, D., Muntean, M., Schaaf, E., Dentener, F., Aardenne, J. A. V., ... & Janssens-Maenhout, G. (2018). Gridded emissions of air pollutants for the period 1970–2012 within EDGAR v4.3.2. Earth System Science Data, 10(4), 1987-2013.

See Also

speciation and read

Examples

```
# load the mapping tables
data(species)
# names of eath mapping tables
for(i in 1:length(names(species)))
    cat(paste0("specie map ",i," ",names(species)[i],"\n"))
# names of species contained in the (first) mapping table
names(species[[1]])
# The first mapping table / species and values
species[1]
```

---

totalEmission *Calculate total emissions*

Description

Caculate the total emission with:

Emission(pollutant) = sum(Vehicles(n) * Km_day_use(n) * Emission_Factor(n,pollutant))

where n is the type of the veicle

Usage

```
totalEmission(v, ef, pol, verbose = TRUE)
```

Arguments

v	dataframe with the vehicle data
ef	emission factor
pol	pollutant name in ef
verbose	display additional information

Value

Return a list with the daily total emission by interest area (cityes, states, countries, etc).

Note

the units (set_units("value",unit) where the recomended unit is g/d) must be used to make the ef data.frame

See Also

rasterSource, lineSource and emission

Examples

```
veic <- vehicles(example = TRUE)

EmissionFactors <- emissionFactor(example = TRUE)

TOTAL <- totalEmission(veic,EmissionFactors,pol = c("CO","PM"))
```

`vehicles` *Tool to set-up vehicle data table*

Description

Return a data frame with 4 columns (vehicle category, type, fuel and avarage kilometers driven) and an aditional column with the number of vehicles for each interest area (cityes, states, countries, etc).

Average daily kilometres driven are defined by vehicle type:

- LDV (Light duty Vehicles) 41 km / day

- TRUCKS (Trucks) 110 km / day

- BUS (Busses) 165 km / day

- MOTO (motorcycles and other vehicles) 140 km / day

The number of vehicles are defined by the distribution of vehicles by vehicle classs and the total number of vehicles by area.

Usage

```
vehicles(
  total_v,
  area_name = names(total_v),
  distribution,
  type,
  category = NA,
  fuel = NA,
  vnames = NA,
  example = FALSE,
  verbose = TRUE
)
```

Arguments

`total_v`	total of vehicles by area (area length)
`area_name`	area names (area length)
`distribution`	distribution of vehicles by vehicle class
`type`	type of vehicle by vehicle class (distribution length)
`category`	category name (distribution length / NA)

`fuel`	fuel type by vehicle class (distribution length / NA)
`vnames`	name of each vehicle class (distribution length / NA)
`example`	a simple example
`verbose`	display additional information

Value

a fleet distribution data.frame for totalEmission function

Note

total_v and area_name must have the same length.

distribution, type, category (if used), fuel (if used) and vnames (if used) must have the same length.

See Also

`areaSource` and `totalEmission`

Examples

```
fleet <- vehicles(example = TRUE)

# or the code bellow for the same result
# DETRAN 2016 data for total number of vehicles for 5 Brazilian states (Sao Paulo,
# Rio de Janeiro, Minas Gerais, Parana and Santa Catarina)
# vahicle distribution of Sao Paulo

fleet <- vehicles(total_v = c(27332101, 6377484, 10277988, 7140439, 4772160),
                  area_name = c("SP", "RJ", "MG", "PR", "SC"),
                  distribution = c( 0.4253, 0.0320, 0.3602, 0.0260,
                                   0.0290, 0.0008, 0.1181, 0.0086),
                  category =  c("LDV_E25","LDV_E100","LDV_F","TRUCKS_B5",
                                "CBUS_B5","MBUS_B5","MOTO_E25","MOTO_F"),
                  type = c("LDV", "LDV", "LDV","TRUCKS",
                          "BUS","BUS","MOTO", "MOTO"),
                  fuel = c("E25", "E100", "FLEX","B5",
                           "B5","B5","E25", "FLEX"),
                 vnames = c("Light duty Vehicles Gasohol","Light Duty Vehicles Ethanol",
                        "Light Duty Vehicles Flex","Diesel trucks","Diesel urban busses",
                             "Diesel intercity busses","Gasohol motorcycles",
                             "Flex motorcycles"))
```

Index