

# Package ‘ukghg’

May 3, 2016

**Title** Greenhouse Gas Fluxes from the UK

**Version** 0.1

**Description** Spatio-temporal predictions of UK GHG emissions.

**Depends** R (>= 3.2.2), mgcv, raster, animation

**Suggests** knitr

**License** Contact the author.

**VignetteBuilder** knitr

**LazyData** true

**RoxygenNote** 5.0.1

## R topics documented:

calcAlpha . . . . .	1
calcFlux . . . . .	2
calcFlux_anthro . . . . .	3
calcFlux_bio . . . . .	3
combineFlux . . . . .	4
unit_conversion . . . . .	5
writeGIF . . . . .	5
writeNetCDF . . . . .	6
<b>Index</b>	7

---

calcAlpha	<i>A calcAlpha Function</i>
-----------	-----------------------------

---

## Description

This function calculates values of alpha, the coefficient.

## Usage

```
calcAlpha(ghgName = c("ch4", "co2", "n2o"), datect, sectorList = 1:10)
```

**Arguments**

ghgName	Greenhouse gas: one of "ch4", "co2", or "n2o". Defaults to "ch4".
datect	A vector of timestamps in POSIXct format.
sectorList	A vector of sector numbers for which alpha values should be returned, e.g. c(1,3,7). Defaults to all.

**Examples**

```
startDate <- as.POSIXct(strptime("01/06/2006", "%d/%m/%Y"), tz = "UTC")
endDate   <- as.POSIXct(strptime("02/06/2006", "%d/%m/%Y"), tz = "UTC")
# create a sequence of dates
nTimes <- 2
datect <- seq(startDate, endDate, length = nTimes)
alpha_df <- calcAlpha("ch4", datect)
```

---

calcFlux	<i>A calcFlux Function</i>
----------	----------------------------

---

**Description**

This function calculates greenhouse gas fluxes from the UK, based on a spatio-temporal model and the national GHG inventory data.

**Usage**

```
calcFlux(ghgName = c("ch4", "co2", "n2o"), datect = datect,
  proj = c("OSGB", "LonLat"), unitType = c("mol", "g"),
  unitSIprefix = c("kilo", "none", "milli", "micro", "nano", "pico"),
  writeNetCDF = TRUE, writeGIF = FALSE, sectorList = 1:10)
```

**Arguments**

ghgName	Greenhouse gas: one of "ch4", "co2", or "n2o". Defaults to "ch4".
datect	A vector of timestamps in POSIXct format.
proj	Geographic projection for the gridded data, either "OSGB" or "LonLat". Defaults to OSGB, LonLat not implemented yet.
unitType	Either molar ("mol") or mass-based ("g").
unitSIprefix	Any standard SI prefix for the output units, from "kilo" to "pico".
writeNetCDF	Write NetCDF output files. Defaults to TRUE.
writeGIF	Produce an animated GIF output file. Requires ImageMagick to be installed. Defaults to FALSE.
sectorList	A vector of sector numbers for which alpha values should be returned, e.g. c(1,3,7). Defaults to all.

**Examples**

```
startDate <- as.POSIXct(strptime("01/06/2006", "%d/%m/%Y"), tz = "UTC")
endDate   <- as.POSIXct(strptime("02/06/2006", "%d/%m/%Y"), tz = "UTC")
# create a sequence of dates
nTimes <- 2
datect <- seq(startDate, endDate, length = nTimes)
myFlux <- calcFlux("ch4", datect, "OSGB", "mol", "nano")
```

---

calcFlux_anthro	<i>A calcFlux_anthro Function</i>
-----------------	-----------------------------------

---

### Description

This function calculates anthropogenic greenhouse gas fluxes from the UK, based on a spatio-temporal model and the national GHG inventory data.

### Usage

```
calcFlux_anthro(ghgName = c("ch4", "co2", "n2o"), datect = datect,
  proj = c("OSGB", "LonLat"), unitType = c("mol", "g"),
  unitSIprefix = c("kilo", "none", "milli", "micro", "nano", "pico"),
  sectorList = 1:10)
```

### Arguments

ghgName	Greenhouse gas: one of "ch4", "co2", or "n2o". Defaults to "ch4".
datect	A vector of timestamps in POSIXct format.
proj	Geographic projection for the gridded data, either "OSGB" or "LonLat". Defaults to OSGB, LonLat not implemented yet.
unitType	Either molar ("mol") or mass-based ("g").
unitSIprefix	Any standard SI prefix for the output units, from "kilo" to "pico".
sectorList	A vector of sector numbers for which alpha values should be returned, e.g. c(1,3,7). Defaults to all.

### Examples

```
startDate <- as.POSIXct(strptime("01/06/2006", "%d/%m/%Y"), tz = "UTC")
endDate <- as.POSIXct(strptime("02/06/2006", "%d/%m/%Y"), tz = "UTC")
# create a sequence of dates
nTimes <- 2
datect <- seq(startDate, endDate, length = nTimes)
myFlux <- calcFlux_anthro("ch4", datect, "OSGB", "mol", "nano")
```

---

calcFlux_bio	<i>A calcFlux_bio Function</i>
--------------	--------------------------------

---

### Description

This function calculates biopogenic greenhouse gas fluxes from the UK, based on a spatio-temporal model and the national GHG inventory data.

### Usage

```
calcFlux_bio(ghgName = c("ch4", "co2", "n2o"), datect = datect,
  proj = c("OSGB", "LonLat"), unitType = c("mol", "g"),
  unitSIprefix = c("kilo", "none", "milli", "micro", "nano", "pico"))
```

**Arguments**

ghgName	Greenhouse gas: one of "ch4", "co2", or "n2o". Defaults to "ch4".
datect	A vector of timestamps in POSIXct format.
proj	Geographic projection for the gridded data, either "OSGB" or "LonLat". Defaults to OSGB, LonLat not implemented yet.
unitType	Either molar ("mol") or mass-based ("g").
unitSIprefix	Any standard SI prefix for the output units, from "kilo" to "pico".

**Examples**

```

startDate <- as.POSIXct(strptime("01/06/2006", "%d/%m/%Y"), tz = "UTC")
endDate   <- as.POSIXct(strptime("02/06/2006", "%d/%m/%Y"), tz = "UTC")
# create a sequence of dates
nTimes <- 2
datect <- seq(startDate, endDate, length = nTimes)
myFlux <- calcFlux_bio("ch4", datect, "OSGB", "mol", "nano")

```

---

combineFlux

*A combineFlux Function*


---

**Description**

This function combines biogenic and anthropogenic greenhouse gas fluxes from the UK, based on a spatio-temporal model and the national GHG inventory data.

**Usage**

```
combineFlux(flux_anthro, flux_bio)
```

**Arguments**

flux_anthro	anthropogenic greenhouse gas fluxes
flux_bio	biogenic greenhouse gas fluxes

**Examples**

```

startDate <- as.POSIXct(strptime("01/06/2006", "%d/%m/%Y"), tz = "UTC")
endDate   <- as.POSIXct(strptime("02/06/2006", "%d/%m/%Y"), tz = "UTC")
# create a sequence of dates
nTimes <- 2
datect <- seq(startDate, endDate, length = nTimes)
flux_anthro <- calcFlux_anthro("ch4", datect, "OSGB", "mol", "nano")
flux_bio    <- calcFlux_bio("ch4", datect, "OSGB", "mol", "nano")
flux_all    <- combineFlux(flux_anthro, flux_bio)

```

---

unit_conversion	<i>A unit_conversion Function</i>
-----------------	-----------------------------------

---

**Description**

This function converts from Tg km-2 y-1 to a standard SI unit.

**Usage**

```
unit_conversion(ghgName = c("ch4", "co2", "n2o"), unitType = c("mol", "g"),
  unitSIprefix = c("kilo", "none", "milli", "micro", "nano", "pico"))
```

**Arguments**

ghgName	Greenhouse gas: one of "ch4", "co2", or "n2o". Defaults to "ch4".
unitType	Either molar ("mol") or mass-based ("g").
unitSIprefix	Any standard SI prefix for the output units, from "kilo" to "pico".

**Examples**

```
unit_conversion("ch4", "mol", "nano")
unit_conversion("co2", "mol", "micro")
unit_conversion("n2o", "mol", "nano")
unit_conversion("ch4", "g", "nano")
```

---

writeGIF	<i>A writeGIF Function</i>
----------	----------------------------

---

**Description**

This function writes an animated GIF output file

**Usage**

```
writeGIF(ghgName, datect, proj, flux, thresh = 20,
  IMconvertPath = "/Progra~1/ImageMagick-6.8.9-Q16/convert.exe")
```

**Arguments**

ghgName	Greenhouse gas: one of "ch4", "co2", or "n2o". Defaults to "ch4".
datect	A vector of timestamps in POSIXct format.
proj	Geographic projection for the gridded data, either "OSGB" or "LonLat". Defaults to OSGB, LonLat not implemented yet.
flux	a ukghg flux object
thresh	Maximum value for z value scale
IMconvertPath	Path to ImageMagick convert executable. Typically "/Progra~1/ImageMagick-6.8.9-Q16/convert.exe" on Windows, depending on version number.

**Examples**

```

startDate <- as.POSIXct(strptime("01/06/2006", "%d/%m/%Y"), tz = "UTC")
endDate   <- as.POSIXct(strptime("02/06/2006", "%d/%m/%Y"), tz = "UTC")
# create a sequence of dates
nTimes <- 2
datect <- seq(startDate, endDate, length = nTimes)
flux_anthro <- calcFlux_anthro("ch4", datect, "OSGB", "mol", "nano")
flux_bio    <- calcFlux_bio("ch4", datect, "OSGB", "mol", "nano")
flux_all    <- combineFlux(flux_anthro, flux_bio)
rf          <- writeGIF("ch4", datect, "OSGB", flux_all, thresh=30)

```

---

writeNetCDF

*A writeNetCDF Function*


---

**Description**

This function writes netCDF output files

**Usage**

```
writeNetCDF(ghgName, datect, proj, flux)
```

**Arguments**

ghgName	Greenhouse gas: one of "ch4", "co2", or "n2o". Defaults to "ch4".
datect	A vector of timestamps in POSIXct format.
proj	Geographic projection for the gridded data, either "OSGB" or "LonLat". Defaults to OSGB, LonLat not implemented yet.
flux	a ukghg flux object

**Examples**

```

startDate <- as.POSIXct(strptime("01/06/2006", "%d/%m/%Y"), tz = "UTC")
endDate   <- as.POSIXct(strptime("02/06/2006", "%d/%m/%Y"), tz = "UTC")
# create a sequence of dates
nTimes <- 2
datect <- seq(startDate, endDate, length = nTimes)
flux_anthro <- calcFlux_anthro("ch4", datect, "OSGB", "mol", "nano")
flux_bio    <- calcFlux_bio("ch4", datect, "OSGB", "mol", "nano")
flux_all    <- combineFlux(flux_anthro, flux_bio)
rf          <- writeNetCDF(flux_all)

```

# Index

## \*Topic **alpha**

calcAlpha, [1](#)

## \*Topic **flux**

calcFlux\_anthro, [3](#)

calcFlux\_bio, [3](#)

combineFlux, [4](#)

writeGIF, [5](#)

writeNetCDF, [6](#)

## \*Topic **units**

calcFlux, [2](#)

unit\_conversion, [5](#)

calcAlpha, [1](#)

calcFlux, [2](#)

calcFlux\_anthro, [3](#)

calcFlux\_bio, [3](#)

combineFlux, [4](#)

unit\_conversion, [5](#)

writeGIF, [5](#)

writeNetCDF, [6](#)