

Package: remindClimateAssessment (via r-universe)

June 10, 2026

Type Package

Title REMIND integration of IIASA's `climate-assessment` package

Version 0.1.1

Date 2026-05-27

Description The REMIND integration of IIASA's `climate-assessment` provides a standardized interface to simple climate models such as MAGICC7.

License LGPL-3

URL <https://github.com/pik-piam/remindClimateAssessment>

Imports dplyr, gdxrrw, lucode2, magrittr, piamInterfaces, purrr, quitte (>= 0.3144.0), readr, rlang, stringr, tidyr, withr, yaml

Suggests testthat

Encoding UTF-8

RoxygenNote 7.3.3

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev git make libharfbuzz-dev libgit2-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

Repository <https://pik-piam.r-universe.dev>

Date/Publication 2026-05-27 14:42:23 UTC

RemoteUrl <https://github.com/pik-piam/remindClimateAssessment>

RemoteRef HEAD

RemoteSha a626d75e500f827b225b524fde470e7d992d9c70

Contents

appendClimateAssessmentToMif	2
archiveClimateAssessmentData	3
assessmentDataToGDX	4

cleanUpClimateAssessment	5
climateAssessmentConfig	6
dumpToGdx	7
emissionDataForClimateAssessment	7
emissionImpulseSenarios	9
exportConfFromYaml	10
renameVariableMagicc7ToRemind	10
renameVariableRemindToMagicc7	11
reportClimateAssessmentConfig	12
runTimeReport	12

Index	14
--------------	-----------

`appendClimateAssessmentToMif`

Append Climate Assessment Data to MIF File

Description

This function appends climate assessment data to a MIF (Model Intercomparison Framework) file. It removes data from old MAGICC7 runs to avoid duplication and then writes the updated data back to the MIF file.

Usage

```
appendClimateAssessmentToMif(qf, mif)
```

Arguments

<code>qf</code>	'quite' data frame containing the climate assessment data to be appended
<code>mif</code>	File path to the MIF to which the data will be appended

Value

The function returns the result of writing the updated data to the MIF file.

`archiveClimateAssessmentData`*Archive Climate Assessment Data*

Description

Archives climate assessment data by copying all *.csv/*.xlsx from climate directory (and, if specified, extra files) to a temporary directory, then tar'ing and gzip'ing the temporary directory & deleting it afterwards

Usage

```
archiveClimateAssessmentData(  
  climateDir,  
  climateArchiveDir = NULL,  
  suffix = "",  
  extraFiles = c(),  
  returnFn = TRUE  
)
```

Arguments

<code>climateDir</code>	Base directory containing climate assessment data
<code>climateArchiveDir</code>	Directory where the climate assessment data will be archived. Default is NULL, in which case a new 'archive' subdir is created in the climate directory. Note: This directory is NOT deleted after archiving, only the temporary directory created within it is
<code>suffix</code>	String appended to the file name. Can be used to identify the climate assessment run mode (i.e. 'report', 'iteration', 'impulse') in the archive file name. Defaults to empty string
<code>extraFiles</code>	Character vector of additional file paths to include in the archive. Default is an empty vector
<code>returnFn</code>	A logical value indicating whether to return the path to the tar file. Defaults to TRUE

Value

If `returnFn` is TRUE, the function returns the path to the tar file. Otherwise, it returns invisible()

Author(s)

Tonn Rüter

Examples

```
## Not run:
cfg <- climateAssessmentConfig("<outputDir>", "iteration")
extraFiles <- c("pm_globalMeanTemperature", "p15_forc_magicc")
archiveClimateAssessmentData(cfg$climateArchiveDir, extraFiles)

## End(Not run)
```

assessmentDataToGDX *Something*

Description

Something

Usage

```
dumpToGDX(
  assessmentData,
  exportConf,
  interpolateYears = NULL,
  logFile = NULL,
  verbose = NULL
)
```

Arguments

assessmentData	Data frame as produced by climate-assessment
exportConf	Associates climate assessment variables with the GDX file names and their GAMS counterpart
interpolateYears	In case the periods in the emission report do not match the one reported by climate assessment, we need to interpolate to be able to integrate them with existing GAMS data
logFile	Path to the log file. Default is NULL
verbose	Determines logging level. If set to TRUE, log messages are printed to std out

Author(s)

Tonn Rüter

Examples

```
## Not run:
# Generates REMIND emission report, extracts relevant variables and reshapes. Note: This function
# can be used in a tidyverse pipeline.
emissionDataForClimateAssessment(
  remind2::reportEmi(fulldata.gdx),
  scenarioName = "SSP2EU-NPi-ar6",
  climateAssessmentYaml = file.path(
    system.file(package = "pamInterfaces"), "iiasaTemplates", "climate_assessment_variables.yaml"
  )
  logfile = "output/missing.log"
)

## End(Not run)
```

cleanUpClimateAssessment

Clean Up Climate Assessment Data

Description

This function cleans up climate assessment data by filtering for specific periods, interpolating missing periods, and renaming variables from MAGICC7 to REMIND naming convention

Usage

```
cleanUpClimateAssessment(qf, periods)
```

Arguments

qf	A 'quitty' object containing the climate assessment data to be cleaned
periods	Vector periods (years) to be considered

Value

A cleaned 'quitty' object with filtered periods, interpolated missing periods, and renamed variables

 climateAssessmentConfig

Configure Climate Assessment

Description

Collects all necessary files and directories to set up climate assessment run from a REMIND output directory

Usage

```
climateAssessmentConfig(outputDir, mode)
```

Arguments

outputDir	A REMIND output directory
mode	Determines which probability files is used. Must be either "report", "iteration" or "impulse"

Value

A list containing the configuration settings for the climate assessment

Examples

```
## Not run:
climateAssessmentConfig("<outputDir>", "iteration") # yields
list(
  outputDir = "/p/tmp/tonnru/remind/output/h_cpol_KLW_d50_2025-02-06_18.10.48",
  scenario = "h_cpol_KLW_d50",
  condaEnv = "/p/projects/rd3mod/python/environments/scm_magicc7_hpc",
  isArchived = FALSE,
  logFile = "<outputDir>/log_climate.txt",
  workersDir = "<outputDir>/climate-assessment-data/workers",
  climateDir = "<outputDir>/climate-assessment-data",
  archiveDir = "",
  scriptsDir = "<Depending on your installation>",
  magiccBin = "<Depending on your installation>",
  variablesFile = "<Depending on piamInterfaces>",
  infillingDatabase = "<Depending on your REMIND default.cfg>",
  probabilisticFile = "<Depending on your REMIND default.cfg>",
  nSets = "<Depending on your REMIND probabilisticFile>",
  ...
)

## End(Not run)
```

dumpToGdx	<i>assessmentDataToGDx</i>
-----------	----------------------------

Description

Something

Usage

```
dumpToGdx(relevantData, fileName, variables)
```

Arguments

relevantData	FOO
fileName	BAR
variables	BAZ

Author(s)

Tonn Rüter

Examples

```
## Not run:
# Generates REMIND emission report, extracts relevant variables and reshapes. Note: This funtion
# can be used in a tidyverse pipeline.
emissionDataForClimateAssessment(
  remind2::reportEmi(fulldata.gdx),
  scenarioName = "SSP2EU-NPi-ar6",
  climateAssessmentYaml = file.path(
    system.file(package = "pamInterfaces"), "iiasaTemplates", "climate_assessment_variables.yaml"
  )
  logFile = "output/missing.log"
)

## End(Not run)
```

emissionDataForClimateAssessment	<i>emissionDataForClimateAssessment</i>
----------------------------------	---

Description

Converts remind emission data from long to wide format suitable for climate assessment. Only considers regions "GLO" and "World" and extracts only the variables needed for climate assessment. Per default these are provided from the AR6 mapping in the pamInterfaces package. The resulting data frame has one column for each year and one row for each variable. For more information visit <https://pyam-iamc.readthedocs.io/en/stable/data.html>

Usage

```
emissionDataForClimateAssessment(
  qf,
  scenario,
  mapping = "AR6",
  variablesFile = NULL,
  logFile = NULL
)
```

Arguments

qf	quite data frame containing the emission data
scenario	Name of the scenario
mapping	Name of the mapping file from the <code>piaInterfaces</code> library, must be 'AR6', 'climateassessment', 'NGFS_AR6' or 'AR6_MAgPIE'. Defaults to 'AR6'
variablesFile	Path to the yaml file containing the variables needed for climate-assessment. If no file path is provided, the function gets the yaml file from the <code>piaInterfaces</code> package
logFile	Path to the log file. Default is "output/missing.log"

Value

quite data frame with the REMIND emission data reshaped for climate assessment

Author(s)

Tonn Rüter

Examples

```
## Not run:
# Generates REMIND emission report, extracts relevant variables and reshapes. Note: This funtion
# can be used in a tidyverse pipeline.
emissionDataForClimateAssessment(
  remind2::reportEmi(fulldata.gdx),
  scenarioName = "SSP2EU-NPi-ar6",
  climateAssessmentYaml = file.path(
    system.file(package = "piaInterfaces"), "iiasaTemplates", "climate_assessment_variables.yaml"
  )
  logFile = "output/missing.log"
)

## End(Not run)
```

```
emissionImpulseScenarios  
  emissionImpulseScenarios
```

Description

Generates emission impulse scenarios from a harmonized and infilled emissions data by repeating the unperturbed scenario as baseline and concatenating perturbed scenarios. Each scenario adds a specified quantity of additional emissions to the specified emission variable in the given time periods. The periods of the original data frame can be extended to observe long term effects in the climate emulation

Usage

```
emissionImpulseScenarios(  
  qf,  
  var = "AR6 climate diagnostics|Infilled|Emissions|CO2|Energy and Industrial Processes",  
  impulse = 3660,  
  periods = c(2020, 2030, 2040, 2050, 2060, 2070, 2080, 2090, 2100, 2110, 2130),  
  endYear = 2200,  
  inbetween = 1  
)
```

Arguments

qf	quitte data frame containing the emission data from a single emission scenario
var	Name of the variable to be perturbed
impulse	Amount of additional emissions. Defaults to 3.66e2, CO2 equivalent in Mt to 1 GtC
periods	Periods in which to add the emissions impulse. Defaults to c(2020, 2030, ..., 2110, 2130)
endYear	Extends the original observation periods. Defaults to 2200
inbetween	Duration of the extended observation periods in years. Defaults to 1

Value

quitte data frame containing all data for the emission impulse scenarios

Author(s)

Tonn Rüter

Examples

```
## Not run:
impulses <- read.quitte(cfg$harmInfEmissionsFile) %>%
  emissionImpulseSenarios() %>%
  write_csv(cfg$emissionsImpulseFile, quote = "none")

## End(Not run)
```

```
exportConfFromYaml      exportConfFromYaml
```

Description

Read yaml export configuration. It determines which variables to extract from the MAGICC7 climate assessment data & to which file they are supposed to be written. Resulting data frame contains magicc7Variable, gamsVariable and fileName columns.

Usage

```
exportConfFromYaml(yamlFileName = "default.yaml")
```

Arguments

yamlFileName Location of the configuration file. Default is 'which_var_in_which_file.yaml'

Value

A tibble data frame containing the association between climate assessment and gams variable names

Author(s)

Tonn Rüter

```
renameVariableMagicc7ToRemind
      renameVariableMagicc7ToRemind
```

Description

Substitutes parts of MAGICC7 variable names to create REMIND compatible variable names

Usage

```
renameVariableMagicc7ToRemind(varName)
```

Arguments

varName Variable to be renamed

Value

Renamed variable

Author(s)

Tonn Rüter

renameVariableRemindToMagicc7
renameVariableRemindToMagicc7

Description

Renames climate assessment variables given in the REMIND convention MAGICC7 compatible variable names

Usage

```
renameVariableRemindToMagicc7(varName, magiccVersion = "7.5.3")
```

Arguments

varName Variable to be renamed

magiccVersion Version string of the MAGICC model in use. Default is "7.5.3"

Value

Renamed variable

Author(s)

Tonn Rüter

```
reportClimateAssessmentConfig
      Report climate assessment configuration
```

Description

Returns a string representation of a climate assessment configuration

Usage

```
reportClimateAssessmentConfig(cfg)
```

Arguments

```
cfg          Path to default.cfg file
```

Value

A string detailing the configuration settings.

Examples

```
## Not run:
cfg <- climateAssessmentConfig("<outputDir>", "iteration")
report <- reportClimateAssessmentConfig(cfg)
cat(report)

## End(Not run)
```

```
runTimeReport      Generate a report of run times.
```

Description

This function takes a named vector of run times and generates a formatted report showing the duration of each run. The names of the run times must follow the pattern 'task_id_1 start' and 'task_id_1 end', where 'task_id_1' can be any string identifier.

Usage

```
runTimeReport(runTimes, prefix = "")
```

Arguments

```
runTimes      A named vector of run times
prefix        A string to prefix each line of the report. Default is "".
```

Value

A formatted string showing the duration of each run.

Examples

```
## Not run:  
# Typical usage: Set up as empty vector  
runTimes <- c()  
# [... run set up ...]  
runTimes <- c("task1 start" = Sys.time())  
# [... run task1 ...]  
runTimes <- c("task1 end" = Sys.time())  
# [... something untimed in between ...]  
runTimes <- c("task2 start" = Sys.time())  
# [... run task2 ...]  
runTimes <- c("task2 end" = Sys.time())  
runTimeReport(runTimes)
```

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

Index

[appendClimateAssessmentToMif](#), [2](#)
[archiveClimateAssessmentData](#), [3](#)
[assessmentDataToGDX](#), [4](#)

[cleanUpClimateAssessment](#), [5](#)
[climateAssessmentConfig](#), [6](#)

[dumpToGDX \(assessmentDataToGDX\)](#), [4](#)
[dumpToGdx](#), [7](#)

[emissionDataForClimateAssessment](#), [7](#)
[emissionImpulseSenarios](#), [9](#)
[exportConfFromYaml](#), [10](#)

[renameVariableMagicc7ToRemind](#), [10](#)
[renameVariableRemindToMagicc7](#), [11](#)
[reportClimateAssessmentConfig](#), [12](#)
[runTimeReport](#), [12](#)