Package ‘airGRteaching’

March 10, 2020

Type Package

Title Teaching Hydrological Modelling with the GR Rainfall-Runoff Models (‘Shiny’ Interface Included)

Version 0.2.8.69

Date 2020-02-28

Depends airGR (>= 1.4.3.52)

Imports dygraphs (>= 1.1.1.6), markdown, plotrix, shiny (>= 1.1.0), shinyjs (>= 1.0), xts

Suggests knitr

Description Add-on package to the ‘airGR’ package that simplifies its use and is aimed at being used for teaching hydrology. The package provides 1) three functions that allow to complete very simply a hydrological modelling exercise 2) plotting functions to help students to explore observed data and to interpret the results of calibration and simulation of the GR (‘Génie rural’) models 3) a ‘Shiny’ graphical interface that allows for displaying the impact of model parameters on hydrographs and models internal variables.

License GPL-2

NeedsCompilation no

URL https://hydrogr.github.io/airGRteaching/

Encoding UTF-8

VignetteBuilder knitr

Author Olivier Delaigue [aut, cre] (<https://orcid.org/0000-0002-7668-8468>), Laurent Coron [aut] (<https://orcid.org/0000-0002-1503-6204>), Pierre Brigode [aut] (<https://orcid.org/0000-0001-8257-0741>), Guillaume Thirel [ctb] (<https://orcid.org/0000-0002-1444-1830>)

Maintainer Olivier Delaigue <airGR@inrae.fr>

Repository CRAN

Date/Publication 2020-03-10 09:20:02 UTC
**R topics documented:**

- airGRteaching ................................................................. 2
- as.data.frame ............................................................... 4
- CalGR ....................................................................... 6
- dyplot ................................................................. 7
- plot ................................................................. 9
- PrepGR .............................................................. 10
- ShinyGR ............................................................ 12
- SimGR .............................................................. 14

**Index**

<table>
<thead>
<tr>
<th>airGRteaching</th>
<th>Tools to Simplify the Use of the airGR Hydrological Package for Education (Including a Shiny Application)</th>
</tr>
</thead>
</table>

**Description**

airGRteaching is an add-on package to the airGR package that simplifies its use and is teaching-oriented. It allows to use with very low programming skills the rainfall-runoff models (GR4H, GR5H, GR4J, GR5J, GR6J, GR2M, GR1A) and a snow melt and accumulation model (CemaNeige). This package also provides graphical devices to help students to explore data and modelling results.

The airGRteaching package has been designed to fulfil a major requirement: facilitating the use of the airGR functionalities by students. The names of the functions and their arguments were chosen to this end.

The package is mostly based on three families of functions:
- the functions that allow to complete very simply a hydrological modelling exercise;
- plotting functions to help students to explore observed data and to interpret the results of calibration and simulation of the GR models;
- a function which runs a 'Shiny' graphical interface that allows for displaying in real-time model parameters impacts on hydrographs.

This package brings into R the hydrological modelling tools developed at INRAE-Antony (Catchment Hydrology research group of the HYCAR Research Unit, France).

# — Modelling Functions

Three functions allow to complete very simply a hydrological modelling exercise:
- preparation of data: PrepGR();
- calibration of the models: CalGR();
- simulation with the models: SimGR()
## — Plotting Functions

airGRteaching provides two types of plotting functions that allow to produce static (`plot()` or dynamic (`dyplot()`) graphics (incl. mouse events and interactive graphics). The devices allow to explore observed data and to interpret the results of calibration and simulation of the GR models.

## — Shiny interface

The package also provides the `ShinyGR()` function, which allows to run a Shiny interface. Thus its is possible to perform:
- interactive flow simulations with parameters modifications;
- automatic calibration;
- display of internal variables evolution;
- time period selection.

A demonstrator of the graphical interface is available for free online on the Sunshine website.

## — Models

The six hydrological models and the snow melt and accumulation model already available in airGR are available in airGRteaching.

These models can be called within airGRteaching using the following model names:
- GR4H: four-parameter hourly lumped hydrological model (Mathevet, 2005)
- GR5H: five-parameter hourly lumped hydrological model (Ficchi, 2017; Ficchi et al., 2019)
- GR4J*: four-parameter daily lumped hydrological model (Perrin et al., 2003)
- GR5J*: five-parameter daily lumped hydrological model (Le Moine, 2008)
- GR6J*: six-parameter daily lumped hydrological model (Pushpalatha et al., 2011)
- GR2M: two-parameter monthly lumped hydrological model (Mouelhi, 2003; Mouelhi et al., 2006a)
- GR1A: one-parameter annual lumped hydrological model (Mouelhi, 2003; Mouelhi et al., 2006b)
- CemaNeige: two-parameter degree-day snow melt and accumulation daily model (combined with GR4H, GR5H, GR4J, GR5J or GR6J) (Valéry et al., 2014)

*: available in the Shiny interface.

For more information and to get started with the package, you can refer to the vignette (`vignette("get_started")`) and go on the airGRteaching website.

## — References


---

**as.data.frame**

*Function to coerce the outputs of PrepGR, CalGR and SimGR to a data.frame*

**Description**

Function to coerce the outputs of PrepGR, CalGR and SimGR to a data.frame

**Usage**

```r
## S3 method for class 'airGRt'
as.data.frame(x, row.names = NULL, ...)
```

**Arguments**

- `x` [object of class `airGRt`] typically an object of class PrepGR, CalGR or SimGR
- `row.names` NULL or a character vector giving the row names for the data.frame. Missing values are not allowed
- `...` additional arguments to be passed to or from methods
Value

data.frame containing:

- **Dates** [POSIXct] vector of dates
- **PotEvap** [numeric] time series of potential evapotranspiration (catchment average) [mm/time step]
- **PrecipObs** [numeric] time series of total precipitation (catchment average) [mm/time step]
- **PrecipFracSolid_CemaNeige** [numeric] time series of solid precipitation fraction (layer average) [-], must be defined if CemaNeige is used
- **TempMeanSim_CemaNeige** [numeric] time series of mean air temperature (layer average) [°C], must be defined if CemaNeige is used
- **Qobs** [numeric] time series of observed flow (for the same time steps than simulated) [mm/time step]
- **Qsim** [numeric] time series of simulated flow (for the same time steps than simulated) [mm/time step] (only for objects of class CalGR or SimGR)

Author(s)

Olivier Delaigue

See Also

- PrepGR, CalGR, SimGR

Examples

```r
library(airGRteaching)

## data.frame of observed data
data(L0123001, package = "airGR")
BasinObs2 <- BasinObs[, c("DatesR", "P", "E", "Qmm", "T")]

## Preparation of observed data for modelling
PREP <- PrepGR(ObsDF = BasinObs2, HydroModel = "GR4J", CemaNeige = FALSE)
head(as.data.frame(PREP))

## Calibration step
CAL <- CalGR(PrepGR = PREP, CalCrit = "KGE2",
             WupPer = NULL, CalPer = c("1990-01-01", "1991-12-31"))
head(as.data.frame(CAL))

## Simulation step using the result of the automatic calibration method to set the model parameters
SIM <- SimGR(PrepGR = PREP, CalGR = CAL, EffCrit = "KGE2",
             WupPer = NULL, SimPer = c("1992-01-01", "1992-12-31"))
head(as.data.frame(SIM))
```
CalGR

Calibration algorithm that optimises the error criterion selected as objective function

Description

Calibration algorithm that optimises the error criterion selected as objective function using the INRAE-HYCAR procedure described by C. Michel

Usage

CalGR(PrepGR, CalCrit = c("NSE", "KGE", "KGE2", "RMSE"),
     WupPer = NULL, CalPer,
     transfo = c("", "sqrt", "log", "inv", "sort"), verbose = TRUE)

Arguments

PrepGR [object of class PrepGR] see PrepGR for details
CalCrit [character] name of the objective function (must be one of "NSE", "KGE", "KGE2" or "RMSE")
WupPer (optional) [character] vector of 2 values to define the beginning and end of the warm-up period ["YYYY-mm-dd" or "YYYY-mm-dd HH:MM:SS"]
CalPer [character] vector of 2 values to define the beginning and end of the calibration period ["YYYY-mm-dd" or "YYYY-mm-dd HH:MM:SS"]
transfo (optional) [character] name of the transformation transformation applied to discharge for calculating the objective function (must be one of ", "sqrt", "log", "inv" or "sort")
verbose (optional) [boolean] logical value indicating if the function is run in verbose mode or not

Value

list object of class CalGR containing:

OptionsCalib [list] object of class RunOptions (see: CreateRunOptions)
Qobs [numeric] series of observed discharges [mm/time step]
OutputsCalib [list] object of class OutputsCalib (see: Calibration)
OutputsModel [list] object of class OutputsModel (see: RunModel)
TypeModel [character] name of the function of the hydrological model used
CalCrit [character] name of the function that computes the error criterion during the calibration step
PeriodModel [list] $WarmUp: vector of 2 POSIXct values defining the beginning and end of the warm-up period, $Run: vector of 2 POSIXct values defining the beginning and end of the calibration period
dyplot

Author(s)
Olivier Delaigue

See Also
CreateRunOptions, CreateInputsCrit, CreateCalibOptions, ErrorCrit_RMSE, ErrorCrit_NSE, ErrorCrit_KGE, ErrorCrit_KGE2, Calibration_Michel

Examples

library(airGRteaching)

## data.frame of observed data
data(L0123001, package = "airGR")
BasinObs2 <- BasinObs[, c("DatesR", "P", "E", "Qmm", "T")]

## Preparation of observed data for modelling
PREP <- PrepGR(ObsDF = BasinObs2, HydroModel = "GR4J", CemaNeige = TRUE)

## Calibration step
CAL <- CalGR(PrepGR = PREP, CalCrit = "KGE2",
WupPer = NULL, CalPer = c("1990-01-01", "1993-12-31"))
str(CAL)

---

dyplot  

Interactive time series plotting of objects of classes PrepGR, CalGR or SimGR.

Usage

## Default S3 method:
dyplot(x, Qsup = NULL, Qsup.name = "Qsup",
col.Precip = c("royalblue", "lightblue"),
col.Q = c("black", "orangered", "grey"), col.na = "lightgrey",
ylab = NULL, main = NULL,
plot.na = TRUE, RangeSelector = TRUE, Roller = FALSE,
LegendShow = c("follow", "auto", "always", "onmouseover", "never"), ...)  

Arguments

x  
[object of class PrepGR, CalGR or SimGR] see PrepGR, CalGR, SimGR for details

Qsup  
(optional) [numeric] additional time series of flows (at the same time step than argument x) [mm/time step]

Qsup.name  
(optional) [character] a label for the legend of Qsup
```r
library(airGRteaching)

## data.frame of observed data
data(L0123001, package = "airGR")
BasinObs2 <- BasinObs[, c("DatesR", "P", "E", "Qmm", "T")]

## Preparation of observed data for modelling
PREP <- PrepGR(ObsDF = BasinObs2, HydroModel = "GR4J", CemaNeige = FALSE)
dyplot(PREP, main = "Observation")

## Calibration step
CAL <- CalGR(PrepGR = PREP, CalCrit = "KGE2",
             WupPer = NULL, CalPer = c("1990-01-01", "1993-12-31"))
dyplot(CAL, main = "Calibration")

## Simulation
```

Author(s)
Olivier Delaigue

See Also

`PrepGR, CalGR, SimGR`

Examples

```
```
plot <- SimGR(PrepGR = PREP, CalGR = CAL, EffCrit = "KGE2",
             WupPer = NULL, SimPer = c("1994-01-01", "1998-12-31"))
dyplot(plot, main = "Simulation")

---

## Description

Time series plotting GR observation objects

## Usage

### S3 method for class 'PrepGR'

plot(x, type = "l",
     col.Precip = "royalblue", col.Q = "black", col.na = "grey",
     xlab = NULL, ylab = NULL, main = NULL,
     plot.na = TRUE, ...)

### S3 method for class 'CalGR'

plot(x, xlab = NULL, ylab = NULL, main = NULL,
     which = c("perf", "iter", "ts"), ...)

### S3 method for class 'SimGR'

plot(x, ...)

## Arguments

- **x**: [object of class PrepGR] see PrepGR for details
- **type**: [character] the type of plot that should be drawn (see plot for details)
- **col.Precip**: (optional) [character] color code or name for precipitation, see par
- **col.Q**: (optional) [character] color code or name for observed flow, see par
- **col.na**: (optional) [character] color code or name for missing values, see par
- **xlab**: (optional) [character] a label for the x-axis (see title)
- **ylab**: (optional) [character] a label for the y-axis (vector of 1 or 2 values for rainfall and flow respectively; see title)
- **main**: (optional) [character] a main title for the plot (see title)
- **plot.na**: [boolean] boolean indicating if the missing values are plotted on the x-axis
- **...**: other parameters to be passed through to plotting functions
- **which**: [character] choice of the plot type ("perf" (default): plot diagnostics; "iter": parameter and calibration criterion values during the iterations of the steepest descent step of the airGR calibration algorithm; "ts": time series of observed precipitation and observed and simulated flows)
PrepGR

Author(s)
Olivier Delaigue

See Also
PrepGR

Examples

library(airGRteaching)

## data.frame of observed data
data(L0123001, package = "airGR")
BasinObs2 <- BasinObs[, c("DatesR", "P", "E", "Qmm", "T")]

## Preparation of observed data for modelling
PREP <- PrepGR(ObsDF = BasinObs2, HydroModel = "GR4J", CemaNeige = FALSE)

## Observed data plotting
plot(PREP)

PrepGR

Creation of the inputs required to run the CalGR and SimGR functions

Description

Creation of the inputs required to run the CalGR and SimGR functions

Usage

PrepGR(ObsDF = NULL, DatesR = NULL, Precip = NULL, PotEvap = NULL,
Qobs = NULL, TempMean = NULL,
ZInputs = NULL, HypsoData = NULL, NLayers = 5,
HydroModel, CemaNeige = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObsDF</td>
<td>(optional) [data.frame] data.frame of dates, total precipitation, potential evapotranspiration, observed discharges and mean air temperature (only if CemaNeige is used) (variables must be in this order; see below for the units)</td>
</tr>
<tr>
<td>DatesR</td>
<td>(optional) [POSIXt] vector of dates required to create the GR and CemaNeige (if used) models inputs. Time zone must be defined as &quot;UTC&quot;</td>
</tr>
<tr>
<td>Precip</td>
<td>(optional) [numeric] time series of total precipitation (catchment average) [mm/time step], required to create the GR and CemaNeige (if used) models inputs</td>
</tr>
<tr>
<td>PotEvap</td>
<td>(optional) [numeric] time series of potential evapotranspiration (catchment average) [mm/time step], required to create the GR model inputs</td>
</tr>
<tr>
<td>Qobs</td>
<td>(optional) [numeric] time series of observed discharges [mm/time step]</td>
</tr>
</tbody>
</table>
TempMean (optional) [numeric] time series of mean air temperature [°C], required to create the CemaNeige model inputs

ZInputs (optional) [numeric] real giving the mean elevation of the Precip and Temp-Mean series (before extrapolation) [m], possibly used to create the CemaNeige (if used) model inputs

HypsoData (optional) [numeric] vector of 101 reals: min, q01 to q99 and max of catchment elevation distribution [m]; if not defined a single elevation is used for CemaNeige (if used)

NLayers (optional) [numeric] integer giving the number of elevation layers requested [-], required to create CemaNeige (if used) model inputs

HydroModel [character] name of the hydrological model (must be one of "GR1A", "GR2M", "GR4J", "GR5J", "GR6J", "GR4H" or "GR5H")

CemaNeige [boolean] option indicating whether CemaNeige should be activated (only available for hourly or daily models, when HydroModel is equal to any of "GR4J", "GR5J", "GR6J", "GR4H" or "GR5H"). See details

Details

If the ObsDF argument is provided, DatesR, Precip, PotEvap, Qobs and TempMean are not necessary, and vice-versa. If one variable is provided in ObsDF and also separately, then only the data included in ObsDF are used.

If the CemaNeige argument is set to TRUE, the default version of CemaNeige is used (i.e. without the Linear Hysteresis, see the details part in CreateRunOptions).

The PrepGR function can be used even if no observed discharges are available. In this case, it is necessary to provide observed discharges time series equal to NA: this means that either the ObsDF observed discharges column or the Qobs arguments, depending on the format of data you provide, must be provided and filled with NAs.

Value

list object of class PrepGR containing the data required to evaluate the model outputs:

InputsModel [list] object of class InputsModel containing the data required to evaluate the model outputs (see: CreateInputsModel outputs)

Qobs [numeric] time series of observed discharges [mm/time step]

HydroModel [character] time series of observed discharges [mm/time step]

Author(s)

Olivier Delaigue

See Also

CreateInputsModel
Examples

library(airGRteaching)

## data.frame of observed data
data(L0123001, package = "airGR")
BasinObs2 <- BasinObs[, c("DatesR", "P", "E", "Qmm", "T")]

## Preparation of observed data for modelling when inputs set by using a data.frame
PREP <- PrepGR(ObsDF = BasinObs2, HydroModel = "GR4J", CemaNeige = FALSE)
str(PREP)

## Preparation of observed data for modelling when inputs set by using independant vectors
PREP <- PrepGR(DatesR = BasinObs2$DatesR, Precip = BasinObs2$P,
    PotEvap = BasinObs2$E, Qobs = BasinObs2$Qmm,
    HydroModel = "GR4J", CemaNeige = FALSE)

## Preparation of observed data for an ungauged catchment (i.e. no observed discharge available)
## Observed discharge set to NA in the Qobs argument
## or in the 4th column of the data.frame if the ObsDF argument is used
PREP <- PrepGR(DatesR = BasinObs2$DatesR, Precip = BasinObs2$P,
    PotEvap = BasinObs2$E, Qobs = NA,
    HydroModel = "GR4J", CemaNeige = FALSE)

ShinyGR

Interactive Web application to run manually the GR4J, GR5 and GR6J hydrological models with or without CemaNeige

Description

Shiny application to understand and to display in a interactive way the impact of each parameter of the GR models on the simulated flows

Usage

ShinyGR(ObsDF = NULL,
    DatesR = NULL, Precip = NULL, PotEvap = NULL, Qobs = NULL, TempMean = NULL,
    ZInputs = NULL, HypsoData = NULL, NLayers = 5,
    SimPer, NamesObsBV = NULL, theme = "RStudio")

Arguments

ObsDF (optional) [data.frame or list of data.frame] data.frame of dates, total precipitation, potential evapotranspiration, observed discharge and mean air temperature (only if CemaNeige is used) (variables must be in this order; see below for the units)

DatesR (optional) [POSIXct] vector of dates required to create the GR and CemaNeige models inputs. Time zone must be defined as "UTC"
Precip (optional) [numeric] time series of total precipitation (catchment average) [mm/time step], required to create the GR and CemaNeige models inputs

PotEvap (optional) [numeric] time series of potential evapotranspiration (catchment average) [mm/time step], required to create the GR model inputs

Qobs (optional) [numeric] time series of observed discharge [mm/time step]

TempMean (optional) [numeric] time series of mean air temperature [°C], required to create the CemaNeige model inputs (if used)

ZInputs (optional) [numeric or list of numerics] real giving the mean elevation of the Precip and TempMean series (before extrapolation) [m], used to create the CemaNeige model inputs (if used)

HypsoData (optional) [numeric or list of numerics] vector of 101 reals: min, q01 to q99 and max of catchment elevation distribution [m]; if not defined a single elevation is used for CemaNeige (if used)

NLayers (optional) [numeric or list of numerics] integer giving the number of elevation layers requested [-], required to create CemaNeige model inputs (if used)

SimPer [character or list of characters] vector of 2 values to define the beginning and the end of the simulation period ["YYYY-mm-dd" or "YYYY-mm-dd HH:MM:SS"], see below for details

NamesObsBV (optional) [character] vector of values to define the data inputs name(s) (if the ObsDF list is not already named)

theme (optional) [character] alternative stylesheet ["RStudio" (default), "Cerulean", "Cyborg", "Flatly", "Inrae", "Saclay", "United" or "Yeti"]

Details

The warm-up period always starts from the first date of the dataset to the time step just before the beginning of the simulation period (SimPer).

The ShinyGR function can be used even if no observed discharges are available. In this case, it is necessary to provide observed discharges time series equal to NA: this means that either the ObsDF observed discharges column or the Qobs arguments, depending on the format of data you provide, must be provided and filled with NAs.

Author(s)

Olivier Delaigue, Laurent Coron, Pierre Brigode

See Also

CalGR, SimGR, plot.CalGR, plot.SimGR

Examples

library(airGRteaching)

## data.frame of observed data of a low-land basin
data(L0123001, package = "airGR")
BV_L0123001 <- BasinObs[0001:6000, c("DatesR", "P", "E", "Qmm", "T")]

```r
```
SimGR

Running one of the GR hydrological models

Description

Function for running the GR hydrological models
Usage

SimGR(PrepGR, CalGR = NULL, Param, EffCrit = c("NSE", "KGE", "KGE2", "RMSE"),
       WupPer = NULL, SimPer,
       transfo = c("", "sqrt", "log", "inv", "sort"), verbose = TRUE)

Arguments

PrepGR [object of class PrepGR] see PrepGR for details
CalGR (deprecated) use the Param argument instead
Param [object of class CalGR or numeric] see CalGR. The length of the vector of pa-

EffCrit [character] name of the efficiency criterion (must be one of "NSE", "KGE", "KGE2"

WupPer (optional) [character] vector of 2 values to define the beginning and end of the

SimPer [character] vector of 2 values to define the beginning and end of the simulation

transfo (optional) [character] name of the transformation applied to discharge for calcu-

verbose (optional) [boolean] logical value indicating if the function is run in verbose

Details

The user can customize the parameters with the Param argument. The user can also use the param-

Value

list object of class SimGR containing:

OptionsSimul [list] object of class RunOptions (see: CreateRunOptions)
OptionsCrit [list] object of class InputsCrit (see: CreateInputsCrit)
OutputsModel [list] object of class OutputsModel (see: RunModel)
Qobs [numeric] series of observed discharges [mm/time step]
TypeModel [character] name of the function of the hydrological model used
CalCrit [character] name of the function that computes the error criterion during the

EffCrit [list] name of the function that computes the error criterion during the simulation

PeriodModel [list] $WarmUp: vector of 2 POSIXct values defining the beginning and end of

the warm-up period; $Run: vector of 2 POSIXct values defining the beginning

and end of the calibration period
Author(s)
Olivier Delaigue

See Also
CreateRunOptions, CreateInputsCrit, RunModel, ErrorCrit_RMSE, ErrorCrit_NSE, ErrorCrit_KGE, ErrorCrit_KGE2

Examples

library(airGRteaching)

## data.frame of observed data
data(L0123001, package = "airGR")
BasinObs2 <- BasinObs[, c("DatesR", "P", "E", "Qmm", "T")]

## Preprocessing of observed data for modeling
PREP <- PrepGR(ObsDF = BasinObs2, HydroModel = "GR4J", CemaNeige = FALSE)

## Calibration step
CAL <- CalGR(PrepGR = PREP, CalCrit = "KGE2",
             WupPer = NULL, CalPer = c("1990-01-01", "1993-12-31"));

## Simulation step using the result of the automatic calibration method to set the model parameters
SIM <- SimGR(PrepGR = PREP, Param = CAL, EffCrit = "KGE2",
             WupPer = NULL, SimPer = c("1994-01-01", "1998-12-31"));

## Simulation step using model parameters set by the user
SIM <- SimGR(PrepGR = PREP, Param = c(270.426, 0.984, 108.853, 2.149), EffCrit = "KGE2",
             WupPer = NULL, SimPer = c("1994-01-01", "1998-12-31"));
str(SIM)
Index

airGRteaching, 2
as.data.frame, 4
CalGR, 2, 4, 5, 6, 7, 8, 13, 15
Calibration, 6
Calibration_Michel, 7
CreateCalibOptions, 7
CreateInputsCrit, 7, 15, 16
CreateInputsModel, 11
CreateRunOptions, 6, 7, 11, 15, 16
dyLegend, 8
dyplot, 3, 7
dyRangeSelector, 8
dyRoller, 8
ErrorCrit_KGE, 7, 16
ErrorCrit_KGE2, 7, 16
ErrorCrit_NSE, 7, 16
ErrorCrit_RMSE, 7, 16
par, 8, 9
plot, 9, 9
plot.CalGR, 13
plot.SimGR, 13
PrepGR, 2, 4–10, 10, 15
rgb, 8
RunModel, 6, 15, 16

ShinyGR, 3, 12
SimGR, 2, 4, 5, 7, 8, 13, 14
title, 9