Package ‘airGRteaching’

October 12, 2022

Type Package
Title Teaching Hydrological Modelling with the GR Rainfall-Runoff Models ('Shiny' Interface Included)
Version 0.2.13
Date 2022-02-03
Depends R (>= 3.6.0), airGR (>= 1.6.9.27)
Imports dygraphs (>= 1.1.1.6), markdown, plotrix, shiny (>= 1.1.0), shinyjs (>= 1.0), xts, graphics, grDevices, stats, utils
Suggests knitr, rmarkdown, testthat, tibble, htmlwidgets (>= 1.5.3)
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/
BugReports https://gitlab.irstea.fr/HYCAR-Hydro/airgrteaching/-/issues
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 2022-03-03 16:40:02 UTC
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 several lumped rainfall-runoff models (GR4H, GR5H, GR4J, GR5J, GR6J, GR2M and 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 user interface that allows for displaying in real-time the impacts of model parameters 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.

# — Graphical user interface

The package also provides the ShinyGR() function, which allows to launch a Shiny interface. Thus it 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 platform.

# — Models

The six hydrological models and the snow melt and accumulation model already available in airGR are accessible from airGRteaching. These models can be called within airGRteaching using the following model names (*: available in the Shiny interface):

- 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)

For more information and to get started with the package, you can refer to the vignette (vignette("get_started", package = "airGRteaching")) 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 'PrepGR'
as.data.frame(x, row.names = NULL, ...)

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

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

- **x**: [PrepGR], [CalGR] or [SimGR] objects
- **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**

```r
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"]. See details
- **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

**Details**

WupPer = NULL indicates that, if available, a period of one year immediately present before the CalPer period is used. In order to disable the warm up of the model, users have to make sure not to provide data before CalPer when preparing the data with PrepGR.
CalGR

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

Author(s)

Olivier Delaigue

See Also

- `airGRteaching plot` and `dyplot` functions to display static and dynamic plots
- `airGR CreateRunOptions,CreateInputsCrit,CreateCalibOptions,ErrorCrit_RMSE,ErrorCrit_NSE, ErrorCrit_KGE,ErrorCrit_KGE2,Calibration_Michel` functions

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 = TRUE)

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

## Structure of CalGR object
str(CAL)

## Parameter and criterion evolution during
## the steepest descent step of the calibration algorithm
plot(CAL, which = "iter")

## Plot diagnostics
```
plot(CAL, which = "perf")

## Static plot of observed and simulated time series
plot(CAL)

## Dynamic plot of observed and simulated time series
dyplot(CAL)

### dyplot

Interactive plots for time series of PrepGR, CalGR and SimGR objects.

#### Description

Interactive plots for time series of PrepGR, CalGR and SimGR objects.

#### Usage

```r
## 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** [PrepGR], [CalGR] or [SimGR] containing the vector of dates (POSIXt) and the time series of numeric values list perturbed inputs and DA model outputs (see PrepGR, CalGR and SimGR)
- **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
- **col.Precip** (optional) [character] vector of 1 (total precip.) or 2 (liquid and solid precip. with CemaNeige) color codes or names for precipitation (these can be of the form "#AABBCC" or "rgb(255, 100, 200)" or "yellow"), see par and rgb
- **col.Q** (optional) [character] vector of up to 3 color codes or names for observed (first value), simulated (second value, if provided) and additional (last value, if provided) flows, respectively (these can be of the form "#AABBCC" or "rgb(255, 100, 200)" or "yellow"), see par and rgb
- **col.na** (optional) [character] color code or name for missing values (these can be of the form "#AABBCC" or "rgb(255, 100, 200)" or "yellow"), see par and rgb
- **ylab** (optional) [character] a label for the y-axis (flow and precipitation)
- **main** (optional) [character] a main title for the plot
- **plot.na** [boolean] indicating if the missing values are plotted on the x-axis
Static plots for time series of PrepGR, CalGR and SimGR objects

Description

Static plots for time series of PrepGR, CalGR and SimGR objects. Also plot of the evolution of parameters and objective function during the calibration step for CalGR object.
Usage

```r
## 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` [PrepGR], [CalGR] or [SimGR] containing the vector of dates (`POSIXt`) and the time series of numeric values list perturbed inputs and DA model outputs (see `PrepGR`, `CalGR` and `SimGR`)
- `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
- `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)
- `...` other parameters to be passed through to plotting functions

Author(s)

Olivier Delaigue

See Also

`airGR plot.OutputsModel` function

`airGR teaching dyplot` function to display dynamic plots

`PrepGR, CalGR, SimGR`
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>
<tr>
<td>TempMean</td>
<td>(optional) [numeric] time series of mean air temperature [°C], required to create the CemaNeige model inputs</td>
</tr>
<tr>
<td>ZInputs</td>
<td>(optional) [numeric] real giving the mean elevation of the Precip and TempMean series (before extrapolation) [m], possibly used to create the CemaNeige (if used) model inputs</td>
</tr>
</tbody>
</table>
**PrepGR**

**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 HydroModel = "GR5H", by default, this model is used without the interception store (i.e. without specifying Imax; see RunModel_GR5H).

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] name of the function of the hydrological model used

**Author(s)**

Olivier Delaigue

**See Also**

airGR teaching plot and dyplot functions to display static and dynamic plots

airGR CreateInputsModel function
**Examples**

```r
library(airGRteaching)

## data.frame of observed data
data(L012301, 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)

## Structure of PrepGR object
str(PREP)

## Static plot of observed time series
plot(PREP)

## Dynamic plot of observed time series
dyplot(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)
## 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 GR2M, GR4J, GR5 and GR6J hydrological models with or without CemaNeige*

**Description**

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

**Usage**

```r
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) [POSIXt] vector of daily or monthly 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", "Intrae", "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.

Several datasets can be proposed at the same time in the interface (see the code example below). A dataset with a daily time step can be proposed at the same time as a dataset at the monthly time step.

CemaNeige can only be used with the daily models at the moment.

**Note**

A demo version of the GUI is available on the Sunshine platform.
Author(s)

Olivier Delaigue, Laurent Coron, Pierre Brigode

See Also

CalGR, SimGR, plot

Examples

library(airGRteaching)

## data.frame of daily observed data of a low-land basin
data(L0123001, package = "airGR")
BV_L0123001 <- BasinObs[0001:6000, c("DatesR", "P", "E", "Qmm", "T")]
BI_L0123001 <- BasinInfo

## data.frame of daily observed data of a mountainous basin
data(L0123002, package = "airGR")
BV_L0123002 <- BasinObs[5000:9999, c("DatesR", "P", "E", "Qmm", "T")]
BI_L0123002 <- BasinInfo

## data.frame of monthly aggregated time series from daily observed data of a low-land basin
BV_L0123001m <- SeriesAggreg(BV_L0123001[BV_L0123001$DatesR < "2000-06-01", ],
                           Format = "%Y%m", ConvertFun = c("sum", "sum", "sum", "mean"))

## Interactive simulation when inputs set by using a data.frame
if (interactive()) {
  ShinyGR(ObsDF = list("Low-land basin" = BV_L0123001, "Mountainous basin" = BV_L0123002),
         ZInputs = list(NULL, median(BI_L0123002$HypsoData)),
         HypsoData = list(NULL, BI_L0123002$HypsoData),
         NLayers = list(5, 5),
         SimPer = list(c("1994-01-01", "1998-12-31"), c("2004-01-01", "2006-12-31")),
         theme = "United")
}

## Interactive simulation using when inputs set by using independant vectors
if (interactive()) {
  ShinyGR(DatesR = BV_L0123002$DatesR,
          Precip = BV_L0123002$P,
          PotEvap = BV_L0123002$E,
          Qobs = BV_L0123002$Qmm,
          Temp = BV_L0123002$T,
          ZInputs = median(BI_L0123002$HypsoData),
          HypsoData = BI_L0123002$HypsoData,
          NLayers = 5,
          SimPer = c("2004-01-01", "2006-12-31"),
          NamesObsBV = "Mountainous basin",
          theme = "Saclay")
}

## Interactive simulation 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
if (interactive()) {
  ShinyGR(DatesR = BV_L0123001$DatesR,
          Precip = BV_L0123001$P,
          PotEvap = BV_L0123001$E,
          Qobs = NA,
          SimPer = c("1994-01-01", "1998-12-31"),
          NamesObsBV = "Low-land basin",
          theme = "Cyborg")
}

## Interactive simulation when inputs are at different time steps
if (interactive()) {
  ShinyGR(ObsDF = list("Low-land basin [daily]" = BV_L0123001,
                   "Low-land basin [monthly]" = BV_L0123001m),
          SimPer = list(c("1994-01-01", "1998-12-01"),
                        c("1994-01-01", "1998-12-01")),
          theme = "Flatly")
}

---

**SimGR**

*Running one of the GR hydrological models*

**Description**

Function for running the GR hydrological models

**Usage**

```r
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 parameters depends on the model used, see below for details
- **EffCrit** [character] name of the efficiency criterion (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"]
- **SimPer** [character] vector of 2 values to define the beginning and end of the simulation period ["YYYY-mm-dd" or "YYYY-mm-dd HH:MM:SS"]
transfo  (optional) [character] name of the transformation applied to discharge for calculating the error criterion (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

Details

The user can customize the parameters with the Param argument. The user can also use the parameters resulting from a calibration. In this case, it is necessary to use the CalGR function.

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

Author(s)

Olivier Delaigue

See Also

airGRteaching plot and dyplot functions to display static and dynamic plots

airGR CreateRunOptions, CreateInputsCrit, RunModel, ErrorCrit_RMSE, ErrorCrit_NSE, ErrorCrit_KGE, ErrorCrit_KGE2 functions

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)
## 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"))

## Structure of SimGR object

str(SIM)

## Plot diagnostics

plot(SIM)

## Static plot of observed and simulated time series

plot(SIM, which = "ts")

## Dynamic plot of observed and simulated time series

dyplot(SIM)
Index

* GR4J
  * airGRteaching, 2

* airGR
  * airGRteaching, 2

* calibration
  * airGRteaching, 2

* efficiency criterion
  * airGRteaching, 2

* hydrology
  * airGRteaching, 2

* model
  * airGRteaching, 2

* shiny
  * airGRteaching, 2

* student
  * airGRteaching, 2

* teaching
  * airGRteaching, 2

airGRteaching, 2

as.data.frame, 4

CalGR, 2, 5, 6, 8–10, 15–17
Calibration, 7
Calibration_Michel, 7
CreateCalibOptions, 7
CreateInputsCrit, 7, 17
CreateInputsModel, 12
CreateRunOptions, 7, 12, 17
dyLegend, 9
dyplot, 3, 7, 8, 10, 12, 17
dyRangeSelector, 9
dyRoller, 9

ErrorCrit_KGE, 7, 17
ErrorCrit_KGE2, 7, 17
ErrorCrit_NSE, 7, 17
ErrorCrit_RMSE, 7, 17

par, 8, 10

plot, 7, 9, 9, 10, 12, 15, 17
plot.OutputsModel, 10
PrepGR, 2, 5, 6, 8–10, 11, 16

rgb, 8
RunModel, 7, 17
RunModel_GR5H, 12

ShinyGR, 3, 13
SimGR, 2, 5, 8–10, 15, 16

title, 10