Package ‘DWBmodelUN’

October 12, 2022

Title Dynamic Water Balance a Hydrological Model

Version 1.0.0

Description A tool to hydrologic modelling using the Budyko framework and the Dynamic Water Balance model with Dynamical Dimension Search algorithm to calibrate the model and analyze the outputs from interactive graphics. It allows to calculate the water availability in basins and also some water fluxes represented by the structure of the model.
See Zhang, L., N., Potter, K., Hickel, Y., Zhang, Q., Shao (2008) <DOI:10.1016/j.jhydrol.2008.07.021> ``Water balance modeling over variable time scales based on the Budyko framework - Model development and testing'', Journal of Hydrology, 360, 117–131.
See Tolson, B., C., Shoemaker (2007) <DOI:10.1029/2005WR004723> ``Dynamically dimensioned search algorithm for computationally efficient watershed model calibration'', Water Resources Research, 43, 1–16.

Depends R (>= 3.4)

Imports dygraphs, htmltools, ncdf4, raster, rgdal

Suggests hydroGOF, knitr, rmarkdown

License GPL-2

BugReports `https://github.com/dazamora/DWBmodelUN/issues`

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Nicolas Duque [aut] (<`https://orcid.org/0000-0002-2631-8573`>),
Carolina Vega [aut] (<`https://orcid.org/0000-0001-7138-0862`>),
Pedro Arboleda [aut] (<`https://orcid.org/0000-0002-7620-825X`>),
David Zamora [aut, cre] (<`https://orcid.org/0000-0002-2256-7054`>),
Camila Garcia [cot] (<`https://orcid.org/0000-0001-8227-7684`>)

Maintainer David Zamora <`dazamoraa@unal.edu.co`>

Repository CRAN

Date/Publication 2020-08-18 08:22:11 UTC

R topics documented:

`basins` *basins*

Description

The polygons of the 23 subbasins across the Sogamoso Basin

Usage

```
basins
```

Format

SpatialPolygonsDataFrame (S4)

basins Shapefile featuring subbasins across the Sogamoso Basin.

References

Duque-Gardeazabal, N. (2018). Estimation of rainfall fields in data scarce colombian watersheds, by blending remote sensed and rain gauge data, using kernel functions. Master thesis. Universidad Nacional de Colombia, Bogotá, Colombia.

`buildGRUmaps` *Build Grouped Response Units in maps*

Description

This function builds raster maps for each parameter based on a raster file where the location of the Grouped Response Units (`GRUs`) are defined. This raster must have the same resolution as the forcing files (i.e., for each cell that is planned to be simulated, there must be forcing time series and a cell assigned to a `GRU`).

Usage

```
buildGRUmaps(gruLoc, parsValues)
```

Arguments

`gruLoc`	raster file that is comprised by numbers from 1 to the total number of `GRUs` that were defined.
`parsValues`	data frame that has the values of the four parameters of each `GRU`. It must have equal number of rows as number of GRU that were defined, and must have four columns which define the `alpha1`, `alpha2`, `d` and `Smax` parameters.

Value

a list which consists of four vectors and four raster, each one of them has the values of a parameter spatialized according with the GRU raster layer.

Author(s)

Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Pedro Felipe Arboleda Obando <pfarboledao@unal.edu.co>
Carolina Vega Viviescas <cvegav@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

Examples

```
data(GRU)
data(param)
gru_maps <- buildGRUmaps(GRU, param)
```

---

cellBasins *Identification of the Cells within a basin*

Description

This function identifies the cells that are within a basin. The runoff produced by those cells will be used, either to calculate the water availability or to compare the simulated variable with the observed runoff in certain streamflow gauges.

Usage

```
cellBasins(gruLoc, basins)
```

Arguments

`gruLoc`	raster file that was used to build GRUs. In this function will be used to number each cell from West to East and from North to South.
`basins`	a shapefile that is comprised each one of the basins where the modeller wants to know the runoff. It must be in the same projection of the gruLoc raster.

Value

a list that comprise two dataframes. The first one, the list of cells in each of the basins contained in the shapefile (`cellBasins`), and second a table that associates the coordinates of each cell with the assigned number (`cellTable`).

Author(s)

Pedro Felipe Arboleda Obando <pfarboledao@unal.edu.co>
Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Carolina Vega Viviescas <cvegav@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

Examples

```
data("GRU","basins")
cellBasins <- cellBasins(GRU, basins)
```

`cells` *cells*

Description

Coordinates (Latitude and Longitude) and ID number of cells in Sogamoso River Basin

Usage

```
cells
```

Format

data.frame

cells Data frame (3 columns by 677 rows), cells coordinates and its ID number.

References

Duque-Gardeazabal, N. (2018). "Estimation of rainfall fields in data scarce colombian watersheds, by blending remote sensed and rain gauge data, using kernel functions". Master thesis. Universidad Nacional de Colombia, Bogotá, Colombia.

`Coord_comparison` *Raster coordinates comparison*

Description

This function compares three characteristics from two rasters: coordinates, resolution, and number of layers (if the rasters have more than one) from two different rasters stacks, and let to know if they are using the same geographical information, or if a new set-up should be done.

Usage

```
Coord_comparison(r1, r2)
```

Arguments

`r1`	raster or data frame. If it is a data frame, it should contain in the first two columns, the X, Y coordinates for every point, in GEOGRAPHIC COORDINATES, the third column and so on should have the variable values, and optionally, the header should have the date, using the format `%m/%Y`.
`r2`	raster or data frame. If it is a data frame, it should contain in the first two columns, the X, Y coordinates for every point, in GEOGRAPHIC COORDINATES, the third column and so on should have the variable values, and optionally, the header should have the date, using the format `%m/%Y`.

Value

It prints on console whether the two rasters are on the same coordinates or not, and return a boolean, TRUE if the rasters are on the same coordinates, and FALSE if not.

Author(s)

Pedro Felipe Arboleda Obando <pfarboledao@unal.edu.co>
Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Carolina Vega Viviescas <cvegav@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

Examples

```
data(P_sogamoso, PET_sogamoso)
Coord_comparison(P_sogamoso, PET_sogamoso)
```

dds — *DDS algorithm to calibrate the model*

Description

This function allows the user to calibrate the DWB or other models with the Dynamical Dimension Search (DDS) algorithm *(Tolson & Shoemaker, 2007)*. As the calibration is performed based on a single value, one should average or create a scalar to evaluate the model's performance. The evaluation can be made using all the streamflow stations, or other variables, between the observed and the simulated values.

Usage

```
dds(xBounds.df, numIter, iniPar = NA, r = 0.2, OBJFUN, ...)
```

Arguments

`xBounds.df`	must be a dataframe which defines the parameter range for searching, with 1st column as the minimum and 2nd column as the maximum of the parameter space.
`numIter`	is an integer that defines the total number of simulations so as to calibrate the model.
`iniPar`	is a vector which contains an optional initial parameter set.
`r`	is a double between 0 and 1 which defines the range of searching in the DDS algorithm, the default value is 0.2.

`OBJFUN`	is a function which returns a scalar value which one is trying to minimize. In this case, the scalar is the Objective Function used to evaluate the model performance.
`...`	other variables and datasets needed to run the model.

Details

dds

Value

`outputs.df` is a four entry list, containing `X_BEST`, `Y_BEST`, `X_TEST` and `Y_TEST`, as they evolve over `numIter` iterations. `X_BEST` and `Y_BEST` are the parameters found by the algorithm, parameters which produce a good value of the **Objective Function** `Y_BEST`. `X_TEST` and `Y_TEST` are the evaluated parameters and their respective performance value.

Author(s)

Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Pedro Felipe Arboleda Obando <pfarboledao@unal.edu.co>
Carolina Vega Viviescas <cvegav@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

References

Tolson, B. A., & Shoemaker, C. A. (2007). "Dynamically dimensioned search algorithm for computationally efficient watershed model calibration". Water Resources Research, 43(1), 1-16.

Examples

```
# Load P and PET databases
data(P_sogamoso, PET_sogamoso)

# Verify that the coordinates of the databases match
Coord_comparison(P_sogamoso, PET_sogamoso)

# Load geographic info of GRU and basins where calibration will be performed
data(GRU,basins)
cellBasins <- cellBasins(GRU, basins)

# Establish the initial modeling conditions
GRU.maps <- buildGRUmaps(GRU, param)
init <- init_state(GRU.maps$smaxR)
g_v <- init$In_ground
s_v <- init$In_storage
rm(init)
```

```
# Load general characteristics of modeling
setup_data <- readSetup(Read = TRUE)
Dates <- seq(as.Date( gsub('[^0-9.]','',colnames(P_sogamoso)[3]),
format = "%Y.%m.%d"),
             as.Date(gsub('[^0-9.]','',tail(colnames(P_sogamoso),1)) ,
             format = "%Y.%m.%d"), by = "month")

# For this calibration exercise, the last date of simulation is
# the same as the final date of calibration
Start.sim <- which(Dates == setup_data[8,1])
End.sim <- which(Dates == setup_data[10,1])
# the first two columns of the P and PET are the coordinates of the cells
Sim.Period <- c(Start.sim:End.sim)+2
Start.cal <- which(Dates == setup_data[9,1])
End.cal <- which(Dates == as.Date("2004-12-01"))
# the first two columns of the P and PET are the coordinates of the cells
Cal.Period <- c(Start.cal:End.cal)+2

#Load observed runoff
data(EscSogObs)

# Function that runs the DWB model
NSE_Sogamoso_DWB <- function(parameters, P, PET, g_v,s_v, Sim.Period, EscObs, Cal.Period){

parameters <- as.vector(parameters)
# Transform the parameters to the format that the model needs
param <- matrix(parameters, nrow = raster::cellStats(GRU,stat="max"))

# Construction of parameter maps from values by GRU
GRU.maps <- buildGRUmaps(GRU, param)
alpha1_v <- GRU.maps$alpha1
alpha2_v <- GRU.maps$alpha2
smax_v <- GRU.maps$smax
d_v <- GRU.maps$d
DWB.sogamoso <- DWBCalculator(P_sogamoso[ ,Sim.Period], PET_sogamoso[ ,Sim.Period],
                            g_v,s_v, alpha1_v, alpha2_v, smax_v,d_v, calibration = TRUE)
Esc.Sogamoso <- varBasins(DWB.sogamoso$q_total, cellBasins$cellBasins)

# model evaluation; in case of possible NA results in the simulation,
# add a conditional assingment to a very high value
sim <- Esc.Sogamoso$varAverage[Cal.Period - 2, ]
obs <- EscSogObs[Cal.Period - 2, ]

if (sum(!is.na(sim)) == prod(dim(sim))){
  numer <- apply((sim - obs)^2, 2, sum, na.rm = TRUE)
  demom <- apply((obs - apply(obs, 2, mean, na.rm = TRUE))^2, 2, sum, na.rm = TRUE)
  nse.cof <- 1 - numer / demom
} else {
  nse.cof <- NA
}

Perf <- (-1)*nse.cof
```

```
if(!is.na(mean(Perf))){
  Mean.Perf <- mean(Perf)
  } else {Mean.Perf <- 1e100}
     return(Mean.Perf)
}

# coupling with the DDS algorithm
xBounds.df <- data.frame(lower = rep(0, times = 40), upper = rep(c(1, 2000), times = c(30, 10)))
result <- dds(xBounds.df=xBounds.df, numIter=2, OBJFUN=NSE_Sogamoso_DWB,
              P=P_sogamoso, PET=PET_sogamoso, g_v=g_v, s_v=s_v, Sim.Period=Sim.Period,
              EscObs=EscSogObs, Cal.Period=Cal.Period)
```

DWBCalculator *DWB model function*

Description

The function performs the distributed DWB hydrological model calculations in the defined domain and time period. It is a model based on the postulates of Budyko, which stated that not only does the actual evapotranspiration depend on potential evapotranspiration, but it is also constrained by water availability *(Budyko, 1974).* The monthly Dynamic Water Balance is underpinned in the demand and supply limits demonstrated by funFU, postulate that is applied to three variables in order to acquire the values of the fluxes and state variables on a monthly time step. The named variables affected by funFU are: the available storage capacity (X), the evapotranspiration opportunity (Y) and the actual evapotranspiration (ET). The model is controlled by four parameters: retention efficiency ($\alpha - 1$), evapotranspiration efficiency ($\alpha - 2$), soil water storage capacity (S_max), and a recession parameter in the groundwater storage that controls the baseflow (d).

Usage

```
DWBCalculator(
  p_v,
  pet_v,
  g_v,
  s_v,
  alpha1_v,
  alpha2_v,
  smax_v,
  d_v,
  calibration = FALSE
)
```

Arguments

p_v	matrix comprised by the precipitation records, that has as rows the number of cells that will be simulated and as columns the number of time steps to be simulated.

pet_v	matrix comprised by the potential evapotranspiration records, that has as rows the number of cells that will be simulated and as columns the number of time steps to be simulated.
g_v	vector comprised of the initial values of the groundwater storage, it must have as many values as cells defined to simulate.
s_v	vector comprised of the initial values of the soil water storage, it must have as many values as cells defined to simulate.
alpha1_v	vector comprised of the values of the retention efficiency that must be between 0 and 1, it must have as many values as cells defined to simulate.
alpha2_v	vector comprised of the values of the evapotranspiration efficiency that must be between 0 and 1, it must have as many values as cells defined to simulate.
smax_v	vector comprised of the values of the soil water storage capacity that must be above 0, it must have as many values as cells defined to simulate.
d_v	vector comprised of the values of the recession constant that must be between 0 and 1, it must have as many values as cells defined to simulate.
calibration	boolean variable which sets the printing of the waitbar that indicates the progress of the calculation of the time series results. The default value is FALSE, indicating that just one run of the model is going to be performed and there is no other waitbar such as the one used by a calibration algorithm.

Details

DWBCalculator only performs one simulation of the distributed hydrological model. The decision to perform other kinds of procedure, such as calibration or assimilation, is entirely on modelers' requirements and necessities. A complementary function is available in the package to calibrate the model using the (dds) algorithm, which has proved to be effective in calibrating models with several GRUs.

To start the model one should set the model features using the readSetup function, load the precipitation and evapotranspiration forcings with the upForcing function, build the GRU and parameter maps with the buildGRUmaps function, compare the coordinates of the uploaded datasets with the Coord_comparison (i.e. the forcings and GRU cells), set the initial conditions of the soil moisture and the groundwater storage, and run the model with DWBCalculator function.

Value

a list comprised by the time series of the hydrological fluxes calculated by the model. The time series have the same length as the forcings that were employed to run the model. The fluxes are:

- q_total a numeric matrix of the total runoff - units (mm/month).
- aet a numeric matrix of actual evapotranspiration - units (mm/month).
- r a numeric matrix of groundwater recharge - units (mm/month).
- qd a numeric matrix of surface runoff - units (mm/month).
- qb a numeric matrix of baseflow - units (mm/month).
- s a numeric matrix of soil water storage - units (mm).
- g a numeric matrix of groundwater storage - units (mm).

Author(s)

Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Pedro Felipe Arboleda Obando <pfarboledao@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>
Carolina Vega Viviescas <cvegav@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

References

Budyko. (1974). "Climate and life". New York: Academic Press, INC.

Zhang, L., Potter, N., Hickel, K., Zhang, Y., & Shao, Q. (2008). "Water balance modeling over variable time scales based on the Budyko framework – Model development and testing". Journal of Hydrology, 360(1-4), 117–131.

Examples

```
# Load P and PET databases
data(P_sogamoso, PET_sogamoso)

# Verify that the coordinates of the databases match
Coord_comparison(P_sogamoso, PET_sogamoso)
# Load geographic info of GRU and parameters per cell
data(GRU, param)
# Construction of parameter maps from values by GRU
GRU.maps <- buildGRUmaps(GRU, param)
alpha1_v <- GRU.maps$alpha1
alpha2_v <- GRU.maps$alpha2
smax_v <- GRU.maps$smax
d_v <- GRU.maps$d

# Establish the initial modeling conditions
init <- init_state(GRU.maps$smaxR)
g_v <- init$In_ground
s_v <- init$In_storage
rm(init)

# Load general characteristics of modeling
setup_data <- readSetup(Read = TRUE)
Dates <- seq(as.Date( gsub('[^0-9.]','',colnames(P_sogamoso)[3]),
format = "%Y.%m.%d"),
             as.Date(gsub('[^0-9.]','',tail(colnames(P_sogamoso),1)) ,
             format = "%Y.%m.%d"), by = "month")
Start.sim <- which(Dates == setup_data[8,1]); End.sim <- which(Dates == setup_data[10,1])
# Sim.Period: the 1st two columns of the P and PET are the coordinates of the cells
Sim.Period <- c(Start.sim:End.sim)+2

# Run DWB model
```

```
DWB.sogamoso <- DWBCalculator(P_sogamoso[ ,Sim.Period],
                    PET_sogamoso[ ,Sim.Period],
                    g_v, s_v, alpha1_v, alpha2_v, smax_v, d_v)
```

EscSogObs *EscSogObs*

Description

Flow rates observed in Sogamoso River Basin at 32 gauges from January 2012 to December 2016

Usage

```
EscSogObs
```

Format

data.frame

EscSogObs Data frame, it contains runoff time series measured at 32 stations within the basin. These gauges belong to the IDEAM monitoring network.

References

Duque-Gardeazabal, N. (2018). "Estimation of rainfall fields in data scarce colombian watersheds, by blending remote sensed and rain gauge data, using kernel functions". Master thesis. Universidad Nacional de Colombia, Bogotá, Colombia.

---

funFU *Fu's function for relationship between precipitation and potential evapotranspiration*

Description

It is a model based on the postulates of Budyko, which stated that not only does the actual evapotranspiration depend on potential evapotranspiration, but it is also constrained by water availability *(Budyko, 1974).*

Usage

```
funFU(PET, P, alpha)
```

Arguments

PET	is the variable which will be inserted as the numerator in Fu's function. It can be a value or a numeric vector, in which case it must have the same length as the denominator vector.
P	is the variable which will be inserted as the numerator in Fu's function. It can be a value or a numeric vector, in which case it must have the same length as the numerator vector.
alpha	parameter of Fu's model which controls the evapotranspiration efficiency, yet it is named depending on the variables used as numerator and denominator. It must be a unique value of type double.

Value

a value or a vector (depending on which kind of data was introduced for numerator and denominator).

Author(s)

Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Pedro Felipe Arboleda Obando <pfarboledao@unal.edu.co>
Carolina Vega Viviescas <cvegav@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

References

Zhang, L., Potter, N., Hickel, K., Zhang, Y., & Shao, Q. (2008). "Water balance modeling over variable time scales based on the Budyko framework – Model development and testing. Journal of Hydrology", 360(1-4), 117–131.

Budyko. (1974). "Climate and life". New York: Academic Press, INC.

Examples

```
PET <- 1000
P <- 2000
alpha <- 0.69  # value used by Zhang et al. (2008)
funFU(PET, P, alpha)
```

graphDWB — *Graph for DWB model results*

Description

This function dynamically graphs the inputs and results of the DWBmodelUN.

Usage

```
graphDWB(var, tp, main, ...)
```

Arguments

`var`	It is a list that contains a time series of type `ts` which you want to graph.
	For (*tp* = *2*), it is recommended to list the simulated runoff series first, followed by the observed.
	For (*tp* = *3*), it must first contain the observed precipitation series, followed by the simulated runoff series and finally the observed runoff.
	For (*tp* = *4*), it must first contain the observed precipitation series, followed by the evapotranspiration series and finally the runoff time series.

`tp`	Variable which is defined to choose the type of graph.
`main`	Main title for the graph.
`...`	Other parameters of the **dygraphs** package.

Details

It has three types of graphs:

- (*tp* = *1*): Plots any variable in a continuous format.
- (*tp* = *2*): Compares the runoff result of the model, with the observations.
- (*tp* = *3*): It allows to show a comparison between the observed and simulated runoff, as well as, with a dataset of precipitation.
- (*tp* = *4*): It presents a comparison between a set of precipitation, actual or potential evapotranspiration and runoff.

Value

Prints a dynamic graph according to the requirements.

Author(s)

Carolina Vega Viviescas <cvegav@unal.edu.co>
Pedro Felipe Arboleda Obando <pfarboledao@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>
Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

Examples

```
# Example 1
data(P_sogamoso)
P.est <- ts(c(t(P_sogamoso[1, -2:-1])), star = c(2001, 1), frequency = 12)
var <- list("Precipitation" = P.est)

graphDWB(var, tp = 1, main = "Precipitation Lat:7.0 Lon:-72.94")

# Example 2
data(simDWB.sogamoso, EscSogObs)
runoff.sim <- ts(simDWB.sogamoso[,25], star = c(2001, 1), frequency = 12)
runoff.obs <- ts(EscSogObs[,25] , star = c(2001, 1), frequency = 12)
var <- list("Runoff.sim" = runoff.sim, "Runoff.obs" = runoff.obs)

graphDWB(var, tp = 2, main = "Runoff at basin closure: Gauge 24067010")

# Example 3
data(P_sogamoso, simDWB.sogamoso, EscSogObs)
P.est <- ts(c(t(P_sogamoso[1, 15:110])), star = c(2002, 1), frequency = 12)
runoff.sim <- ts(simDWB.sogamoso[13:108 ,25], star = c(2002, 1), frequency = 12)
runoff.obs <- ts(EscSogObs[13:108 ,25] , star = c(2002, 1), frequency = 12)
var <- list("Precipitation" = P.est,"Runoff.sim" = runoff.sim, "Runoff.obs" = runoff.obs)

graphDWB(var, tp = 3, main = "DWB results at Sogamoso Basin closure point")

# Example 4
data(P_sogamoso, PET_sogamoso, simDWB.sogamoso)
P <- ts(c(t(P_sogamoso[1, -2:-1])), star = c(2001, 1), frequency = 12)
PET <- ts(c(t(PET_sogamoso[1, -2:-1])), star = c(2001, 1), frequency = 12)
runoff.sim <- ts(simDWB.sogamoso[ ,25], star = c(2001, 1), frequency = 12)
var <- list("P" = P,"PET" = PET, "Runoff.sim" = runoff.sim)

graphDWB(var, tp = 4, main = "General Comparison Sogamoso Basin")
```

GRU *GRU*

Description

Raster data of Group Response Units in Sogamoso River Basin

Usage

```
GRU
```

Format

RasterLayer

GRU Raster, it represents the ten (10) Group Response Units across the Sogamoso River Basin.

`gru_maps` *gru_maps*

Description

Spatial distribution of DWB model parameters in Sogamoso River basin

Usage

```
gru_maps
```

Format

list and raster

alpha1 a vector with alpha1 values for each GRU

alpha2 a vector with alpha2 values for each GRU

smax a vector with Smax values for each GRU

d a vector with d values for each GRU

alpha1R a raster with alpha1 values for each GRU

alpha2R a raster with alpha2 values for each GRU

smaxR a raster with Smax values for each GRU

dR a raster with d values for each GRU init_state *Initial conditions of the model*

Description

This function uploads or creates the initial conditions of the two-state variables present in the DWB model, in raster format.

Usage

```
init_state(raster)
```

Arguments

raster	It could be a raster containing the maximum storage in the root zone or two raster with the initial conditions of storage

Details

It requires the raster composed of the Srmax values that were created using the buildGRUmaps function or two rasters previously created with the initial conditions of the soil water and groundwater storage. If there is only be one raster found, the function creates those two rasters using the value of the provide raster reduced by half.

Value

A list containing initial conditions in storage and in ground.

Author(s)

Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Pedro Felipe Arboleda <pfarboledao@unal.edu.co>
Carolina Vega Viviescas <cvegav@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - Sede Bogota

References

Budyko. (1974). "Climate and life". New York: Academic Press, INC.

Zhang, L., Potter, N., Hickel, K., Zhang, Y., & Shao, Q. (2008). "Water balance modeling over variable time scales based on the Budyko framework - Model development and testing. Journal of Hydrology", 360(1-4), 117-131.

Examples

```
library(raster)

# Example 1
data(gru_maps)
init <- init_state(gru_maps$smaxR)

# Example 2
data(In_storage, In_ground)
init <- init_state(stack(In_storage, In_ground))
```

In_ground *In_ground*

Description

Initial conditions in the soil storage in DWB model of Sogamoso River Basin

Usage

```
In_ground
```

Format

RasterLayer

In_ground Raster, initial conditions in the soil storage.

In_storage *In_storage*

Description

Initial conditions in the groundwater storage in DWB model of Sogamoso River Basin

Usage

```
In_storage
```

Format

RasterLayer

In_storage Raster, initial conditions in the groundwater storage.

`param` *param*

Description

Values to four parameters ($\alpha - 1$, $\alpha - 2$, `d`, `S_max`) of DWB model in each GRU

Usage

```
param
```

Format

data.frame

param Data frame, it should represent a GRU in each row, and parameter values in each column. GRU rank should match the GRU number used in the GRU raster.

`PET_sogamoso` *PET_sogamoso*

Description

Distributed monthly potential evapotranspiration in Sogamoso River Basin from January 2012 to December 2016

Usage

```
PET_sogamoso
```

Format

data.frame

PET_sogamoso Data frame, it contains evapotranspiration data, representing the cells in each row, and the evapotranspiration info. by month in each column. The cell rank should match the cell ID in the cells data frame.

References

Duque-Gardeazabal, N. (2018). "Estimation of rainfall fields in data scarce colombian watersheds, by blending remote sensed and rain gauge data, using kernel functions". Master thesis. Universidad Nacional de Colombia, Bogotá, Colombia.

printVar — *Print or write variables of interest*

Description

This function allows to print or write some of the variables simulated by the DWB model.

Usage

```
printVar(variable, coor_cells, var_name, coord_sys, dates, as, path_var = "")
```

Arguments

`variable`	corresponds to the results of a specific variable of the DWBCalculator.
`coor_cells`	coordinates of the cells in the same order that were simulated and that will be used to create the results in raster format, this is done from the data frames which contain the simulated results
`var_name`	name of the variable that will be printed (e.g., q_total, aet, r, qd, qb, s, g)
`coord_sys`	geographic or projected coordinate system.
`dates`	dates that were simulated.
`as`	option to print the results as independent 'raster' (`.tif`) or in a 'NetCDF' file (`.nc`).
`path_var`	path of the directory where one wants to print the files

Value

It saves in a folder previously created a set of raster files with the results of the variable of interest.

Author(s)

Carolina Vega Viviescas <cvegav@unal.edu.co>
Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Pedro Felipe Arboleda Obando <pfarboledao@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

Examples

```
data(sogamoso)
dwb_results <- sogamoso$dwb_results
data(cells)
dates <- seq(as.Date("2001-01-01"), as.Date("2010-12-01"), by="month")
coord_sys <- "+init=epsg:4326"
r <- dwb_results[[3]][,1:20]
```

```
printVar(r, cells, var_name = "r", coord_sys, dates, as = "NetCDF", path_var = tempdir())
```

`P_sogamoso` *P_sogamoso*

Description

Distributed monthly precipitation in Sogamoso River Basin from January 2012 to December 2016

Usage

```
P_sogamoso
```

Format

data.frame

P_sogamoso Data frame, it should represent the cells in each row, and the precipitation info. by month in each column. The cell rank should match the cell ID in the cells data frame.

References

Duque-Gardeazabal, N. (2018). "Estimation of rainfall fields in data scarce colombian watersheds, by blending remote sensed and rain gauge data, using kernel functions". Master thesis. Universidad Nacional de Colombia, Bogotá, Colombia.

`r.cells` *r.cells*

Description

Sogamoso basin raster, which lists the number of total cells

Usage

```
r.cells
```

Format

RasterLayer

r.cells Raster, data frame Cells converted to raster format.

`readSetup` *Read the model setup*

Description

This function reads the setup features of the model. These include the dates that define the simulated time period, and also the variables that will be printed in individual directories. It reads the information from a `RData` file, and returns the identified variables in a tailored dataframe. Optionally, one can insert the string setup in a dataframe and hence use it in this function.

Usage

```
readSetup(Read = TRUE, setup)
```

Arguments

`Read` is a boolean which is used to identify whether the modeller has created its own dataframe in R or the setup is read from file previously written in the `data` directory. An example of the file is contained in the package. The default value is `TRUE`, meaning that it will read the plain file from the `data` directory.

`setup` is an optional dataframe that contains the character strings which specifies dates and variables to be printed. The first seven rows must be character strings specifying the actions regarding if the modeller requires to print the simulated variables. The order is: calibration mode, print variables, print total runoff, print soil moisture, print actual ET, print direct runoff, print baseflow. Those strings must be *YES* or *NO*. The next four rows are: the initial date of simulation, the initial date for calibration, the final date of simulation, and the final date of calibration. This dates must be in the format year-month-day. If calibration is not required, those dates are ignored.

Value

An organized dataframe which defines the model setup.

Author(s)

Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Pedro Felipe Arboleda Obando <pfarboledao@unal.edu.co>
Carolina Vega Viviescas <cvegav@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

Examples

```
setup <- readSetup(Read = TRUE)  # run if you would like to upload the example setup

data(setup_data)
setup <- readSetup(Read = TRUE, setup_data)

# Create your own setup
a <- rep("no",7)
b <- "1990-01-01"
c <- "1991-01-01"
d <- "2012-12-15"
e <- "2012-12-10"
table_setup <- data.frame(set=a,stringsAsFactors = FALSE)
table_setup <- rbind(table_setup, b, c, d, e)
setup <- readSetup(Read = FALSE, table_setup)
```

`setup_data` *setup_data*

Description

Data.frame with the initial configuration of the model run

Usage

```
setup_data
```

Format

data.frame

setup_data Data frame, contains the set-up and print options to run the DWBmodelUN. It consists of 11 parameters: the first seven are configurations of orders whose values can be 'yes' or 'no', indicating 1) If the model must be calibrated, 2) If the variables must be saved in raster format and which variables 3) R - Total runoff , 4) S - Soil moisture storage, 5) AET - Actual evapotranspiration, 6) Qd - Surface runoff , and 7) Qb - Base flow. The last four variables are dates and refer to the times of the input series, and the start and end times of the simulation and calibration of the model.

`simDWB.sogamoso` *simDWB.sogamoso*

Description

Simulated runoff by the DWBmodelUN in the same stations where there were observed data from the Sogamoso basin

Usage

```
simDWB.sogamoso
```

Format

data.frame

simDWB.sogamoso Data frame, it contains simulated runoff time series at the same 32 stations within the basin.

References

Duque-Gardeazabal, N. (2018). "Estimation of rainfall fields in data scarce colombian watersheds, by blending remote sensed and rain gauge data, using kernel functions". Master thesis. Universidad Nacional de Colombia, Bogotá, Colombia.

`sogamoso` *Sogamoso River Basin data*

Description

Sogamoso River Basin data

Usage

```
sogamoso
```

Format

The list contains

- basins:Shapefile featuring subbasins across the Sogamoso Basin.
- cells:Data frame (3 columns by 677 rows), cells coordinates and its ID number.
- dwb_results:List, it contains the DWB model's outputs in matrix format, q_total- total runoff, aet- actual evapotranspiration, r- recharge, qd- Surface runoff, qd- baseflow, s- soil storage, g- groundwater storage.

- GRU:Raster, it represents the ten (10) Grouped Response Units across the Sogamoso River Basin.
- In_ground:Raster, initial conditions in the groundwater storage.
- In_storage:Raster, initial conditions in the soil storage.
- P_sogamoso:Data frame, it should represent the cells in each row, and the precipitation info. by month in each column. The cell rank should match the cell ID in the cells data frame. The first two columns are the coordinates of the cells
- param:Data frame, it should represent a GRU in each row, and parameter values in each column. GRU rank should match the GRU number used in the GRU raster.
- PET_sogamoso:Data frame, it contains evapotranspiration data, representing the cells in each row, and the evapotranspiration info. by month in each column. The cell rank should match the cell ID in the cells data frame.The first two columns are the coordinates of the cells
- r.cells:Raster, data frame Cells converted to raster format.
- setup_data:Data frame, it contains the set-up and printing options to run the DWBmodelUN. It consists of 11 parameters: the first seven are configurations of orders whose values can be 'yes' or 'no', indicating 1) If the model must be calibrated, 2) If the variables must be saved in raster format and which variables 3) R - Total runoff , 4) S - Soil moisture storage, 5) AET - Actual evapotranspiration, 6) Qd - Surface runoff , and 7) Qb - Base flow. The last four variables are dates and refer to the times of the input series, and the start and end times of the simulation and calibration of the model.
- EscSogObs:Data frame, it contains runoff time series measured at 32 stations within the basin. These gauges belong to the IDEAM monitoring network.
- simDWB.sogamoso:Data frame, it contains simulated runoff time series at the same 32 stations within the basin.

`upForcing` *Upload Forcings*

Description

This function loads the precipitation and evapotranspiration estimates that will be used to run or force the DWB model (`DWBCalculator`). If files are in raster format, it saves a variable cointaining the inputs in table format.

Usage

```
upForcing(
  path_p = tempdir(),
  path_pet = tempdir(),
  file_type = "raster",
  format = "GTiff"
)
```

Arguments

`path_p`	is a character string that specifies the directory where the precipitation rasters or the csv file are stored. The csv file must have nrows = N° of cells and ncol= N° of time steps.
`path_pet`	is a character string that specifies the location of the potential evapotranspiration rasters or the csv file are stored. The csv file must have nrows= N° of cells and ncol= N° of time steps.
`file_type`	Character string that specifies the forcing file formats, it should be "raster" or "csv", the default value is "raster".
`format`	Character string that specifies the format file of the Rasters, possible values are "GTiff" and "NetCDF". Default value is "GTiff".

Details

The character strings that control the location of the forcing files are as default "*./precip/*" and "*./pet/*" for precipitation and potential evapotranspiration, but can be change to other directories. However, if one's intention is to upload them from NetCDF files, the **strings must be completely changed** to a complete path that includes the name and extension of the file.

Value

a list containing the two objects (P and PET).

Author(s)

Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Pedro Felipe Arboleda <pfarboledao@unal.edu.co>
Carolina Vega Viviescas <cvegav@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

---

`varBasins` *value of a variable in each subbasin*

Description

This function retrieves the value of a variable in each of the cells that are within a basin boundary. It also returns the average time series value of the variable.

Usage

```
varBasins(var, cellBasins)
```

Arguments

`var`	one of the dataframe results returned from the DWBcalculator function
`cellBasins`	first entry of the cellBasins function that consists of a list of vectors. Each one of the vectors contains the cell numbers of each basin

Value

a list of two elements. The first one is the time series average value of the variable, and the second is a list of dataframes each one of them contains the time series of each of the cells that are within a basin

Author(s)

Nicolas Duque Gardeazabal <nduqueg@unal.edu.co>
Pedro Felipe Arboleda Obando <pfarboledao@unal.edu.co>
Carolina Vega Viviescas <cvegav@unal.edu.co>
David Zamora <dazamoraa@unal.edu.co>

Water Resources Engineering Research Group - GIREH Universidad Nacional de Colombia - sede Bogota

Examples

```
data(sogamoso,GRU,basins)
dwb_results <- sogamoso$dwb_results
Run <- dwb_results$q_total
cellBasins <- cellBasins(GRU, basins)
cellBasins <- cellBasins$cellBasins

Runoff.Sogamoso <- varBasins(Run, cellBasins)
```

Index