

Package ‘ensemblehydromodel’

January 18, 2022

Title What the Package Does (One Line, Title Case)

Version 0.0.0.9000

Description What the package does (one paragraph).

License ‘use_mit_license()’, ‘use_gpl3_license()’ or friends to pick a license

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports magrittr, airGR, hydroGOF

R topics documented:

calc_hydroGOF	2
calc_subseasonal_validation_results	2
calc_validation_results	4
calibrate_model	5
call_cal_fn	6
create_input	7
default_cal_par	8
find_monthly_indices	8
get_family	9
hydro_family	9
load_meteo_data	10
norm_minmax	10
optim_fn	11
save_airGR_plot	13
save_cal_val_plot	13
set_airGR_par	14
simulate_model	15
split_data_set	15
transfo_param	16
transfo_q	17
validate_model	18
write_ascii	18
Index	20

calc_hydroGOF	Wrapper function around hydroGOF functions
---------------	--

Description

Calculates Goodness-of-Fit functions for two runoff series

Usage

```
calc_hydroGOF(GOF_fun, Qsim, Qobs, na.rm = TRUE)
```

Arguments

GOF_fun	a function, or (todo consider only a functional) a string with function name, of the format GOF_fun(Qsim,Qobs,na.rm = "TRUE"), typically from the hydroGOF package
Qsim	vector, matrix, data.frame etc of simulated runoff values
Qobs	vector, matrix, data.frame etc of observed runoff values
na.rm	a logical value indicating if NA should be removed

Value

transformed runoff in same format as input

See Also

[hydroGOF](#)

Examples

```
# for the first example the function KGE must be loaded,
# for example with library(hydroGOF)
calc_hydroGOF(KGE, 1:10, seq(0, 9))
# this is NA
calc_hydroGOF("KGE", 1:10, rep(0, 10))
# this is also NA
calc_hydroGOF(KGE, 1:10, as.numeric(rep(NA, 10)))
```

calc_subseasonal_validation_results	<i>Calculates subseasonal validation results</i>
-------------------------------------	--

Description

Within a subset of hydrological input data hydro_data subset with ind it calculates performances metrics for monthly defined periods by calling [calc_validation_results](#) with col_name = "season" and the period names from the names in val_subseason. Returns a new data frame with an additional column called col_name with entries period_name

Usage

```
calc_subseasonal_validation_results(
  val_subseason,
  dates,
  ind,
  period_name,
  col_name = "period",
  Qsim,
  Qobs,
  val_crit_transfo = "KGE_none"
)
```

Arguments

val_subseason	a list with named arrays of two digits describing months used to calculate sub-seasonal validation metrics
ind	indices used for subsetting hydro_data
period_name	entries in column col_name naming the subset period
col_name	additional column name in returned data frame (default is period)
Qsim	vector with simulated runoff
Qobs	vector with observed runoff
val_crit_transfo	a vector of strings specifying validation criteria and a runoff transformation separated by a "_". supported are validation criteria from the hydroGOF package usable by the calc_hydroGOF function, for supported runoff transformations please refer to transfo_q
date	date a vector of dates (e.g. Dates, Posix) with load_meteo_data

Value

a data frame as [validate_model](#) but with two additional columns naming the period and the season

See Also

[calc_validation_results](#), [validate_model](#), [calc_hydroGOF](#), [transfo_q](#)

Examples

```
perf_cal <- calc_subseasonal_validation_results(
  val_subseason = list(
    spring = c("02", "03", "04", "05"),
    summer = c("06", "07", "08", "09"),
    hydro_data$BasinObs$DatesR,
    split_indices$ind_cal, "calibration",
    col_name = "period",
    simulation_results$Qsim, Qobs,
    val_crit_transfo =
      c(
        "KGE_none", "NSE_none", "VE_none", "pbias_none",
        "KGE_inv", "NSE_inv",
        "KGE_sqrt", "NSE_sqrt"
      )
  )
)
```

```
)
)
```

calc_validation_results

Calculates validation results for a subset period

Description

Helper function which calls [validate_model](#) and subsets Qsim and Qobs with ind, and returns a new data frame with an additional column called col_name with entries period_name

Usage

```
calc_validation_results(
  ind,
  period_name,
  col_name = "period",
  Qsim,
  Qobs,
  val_crit_transfo = "KGE_none"
)
```

Arguments

ind	indices used for subsetting Qsim and Qobs
period_name	entries in column col_name naming the subset period
col_name	additional column name in returned data frame (default is period)
Qsim	vector with simulated runoff
Qobs	vector with observed runoff
val_crit_transfo	a vector of strings specifying validation criteria and a runoff transformation separated by a "_". supported are validation criteria from the hydroGOF package usable by the calc_hydroGOF function, for supported runoff transformations please refer to transfo_q

Value

a data frame as [validate_model](#) but with an additional columns naming the period

See Also

[validate_model](#), [calc_hydroGOF](#), [transfo_q](#)

Examples

```
calc_validation_results(1:5, "summer", "season", 1:10, seq(0, 9))
```

calibrate_model	<i>Model calibration</i>
-----------------	--------------------------

Description

Calibrates a hydrological model

Usage

```
calibrate_model(
  hydro_data,
  split_indices,
  model,
  input,
  error_crit_transfo = "KGE_none",
  cal_maximize = TRUE,
  cal_fn = "DEoptim",
  do_transfo_param = FALSE,
  cal_par = default_cal_par
)
```

Arguments

split_indices	a list of indices from split_data_set containing the elements ind_cal and ind_warm
model	a string specifying the hydrological model, currently implemented are airGR and TUWmodel package models
input	the output of the create_input function, dependent on the model choice, in general containing information about date, precipitation air temperature and potential evapotranspiration and spatial information as area of the catchment or hypsometric curves
error_crit_transfo	string specifying the error criterion for calibration and the runoff transformation separated by a "_" supported are validation criteria from the hydroGOF package usable by the calc_hydroGOF function, for supported runoff transformations please refer to transfo_q
cal_maximize	a logical indicating if the calibration error criterion should be maximized (or minimized)
cal_fn	a string specifying optimization algorithm to be used for the model calibration. Currently Calibration_Michel is implemented for airGR models, and DEoptim , hydroPSO and malschains are supported for all models.
do_transfo_param	logical indicating if parameter transformation to a hypercube should be applied during calibration
cal_par	a list of calibration settings dependent on the calibration function. Package default values default_cal_par can be overwritten by users.
BasinObs	data frame with time series of input data from load_meteo_data

Value

a list of calibration results with elements called

- model_param a vector of calibrated model parameters,
- error_crit_transfo the error criterion and transformation used,
- error_crit_val the value of the error criterion,
- more info calibration function and model specific output information.

Note

Calibration_Michel does not call `optim_fn`, this is why the large list of otherwise global variables is still required as input todo: 1) check if some input is really needed for Calibration_Michel as e.g. BasinObs 2) think about how to easily access and change cal_par values for end user

See Also

`call_cal_fn`, `optim_fn`

Examples

```
calibration_results <- calibrate_model(
  hydro_data, split_indices, model, input, error_crit_transfo, cal_maximize,
  cal_fn, do_transfo_param, cal_par
)
```

call_cal_fn

Call calibration function

Description

Call several optimization functions for model calibration which are available as R packages

Usage

```
call_cal_fn(
  cal_fn,
  hydro_data,
  split_indices,
  model,
  input,
  error_crit,
  cal_maximize,
  cal_q_transfo,
  do_transfo_param,
  cal_par = default_cal_par
)
```

Arguments

cal_fn	a string specifying optimization algorithm to be used for the model calibration. Currently "DEoptim", "hydroPSO" and "malschains" are supported.
hydro_data	a list or data frame containing observed runoff loaded with load_meteo_data
split_indices	indices indicating the warm-up and calibration period, i.e. the output of split_data_set
model	a string indicating the hydrological model
input	model input returned by create_input
error_crit	a string of a function indicating an error criterion, e.g. KGE , or others from the hydroGOF which are usable by calc_hydroGOF functions
cal_maximize	a logical indicating if function should be maximized
cal_q_transfo	a string indicating how runoff should be transformed with transfo_q
do_transfo_param	logical indicating whether calibration is done with real or transformed parameters
cal_par	a list of calibration settings dependent on the calibration function. Package default values default_cal_par can be overwritten by users.

Examples

```
cal_output <- call_cal_fn(
  cal_fn, hydro_data, split_indices, model, input,
  error_crit, cal_maximize, cal_q_transfo, do_transfo_param, cal_par
)
```

create_input	<i>Create model input</i>
--------------	---------------------------

Description

Create input structure dependent on hydrological model choice

Usage

```
create_input(model, BasinObs, BasinInfo)
```

Arguments

model	a string indicating a hydrological model
BasinObs	data frame with time series of input data from load_meteo_data
BasinInfo	a list with spatial basin information

Value

model a string specifying the hydrological model

Examples

```
create_input("TUW", BasinObs, BasinInfo)
```

default_cal_par	<i>Default calibration parameters</i>
-----------------	---------------------------------------

Description

Default calibration parameters

Usage

```
default_cal_par
```

Format

An object of class list of length 7.

find_monthly_indices	<i>Monthly indices</i>
----------------------	------------------------

Description

Returns monthly indices for specified months within a date vector

Usage

```
find_monthly_indices(date, months, ind = seq_along(date))
```

Arguments

date	a vector of dates (e.g. Dates, Posix)
months	a vector of months in double digits strings, e.g. c("02", "03")
ind	a vector of indices which can be used to subset the time dependent elements in input.

Value

a vector with indices

Examples

```
find_monthly_indices(seq(from = as.Date("1981-01-01"), to = as.Date("2020-12-31"), by = "days"), c("02", "03"))
```

get_family	<i>Get the family of a model</i>
------------	----------------------------------

Description

Reverse lookup table to get the package/family of a hydrological model

Usage

```
get_family(model_str)
```

Arguments

model_str a string indicating a hydrological model

Value

the package or family name

Examples

```
get_family("GR4J")
```

hydro_family	<i>Hydrofamily or Package</i>
--------------	-------------------------------

Description

Hydrofamily or Package

Usage

```
hydro_family
```

Format

An object of class `data.frame` with 7 rows and 2 columns.

load_meteo_data	<i>Load meteo data</i>
-----------------	------------------------

Description

Loads a data frame stored in an .rds file containing meteo data as required by the airGR family (see https://odelaigue.github.io/airGR/page_1_get_started.html). Required are columns listed under Details

Usage

```
load_meteo_data(file, tzone = "UTC")
```

Arguments

file	file
tzone	transfers the DatesR column to POSIXt to time zone tzone if "UTC" (default) is chosen, a Date vector is not transferred to another time zone

Details

Required columns:

- DatesR: dates in the POSIXt or Date format
- P: average precipitation [mm/day]
- T: catchment average air temperature [<U+2103>]
- E: catchment average potential evapotranspiration [mm/day]
- Qmm: outlet discharge [mm/day]

Value

a data frame BasinObs

Examples

```
load_meteo_data("D:/input/airGR/HSU_2044.rds")
```

norm_minmax	<i>Min-max normalization and vice versa</i>
-------------	---

Description

Scales data to [0,1] with min and max of applied data or with specified bounds

Usage

```
norm_minmax(x, min, max, direction = "RT")
```

Arguments

x	data to be scaled, can be a vector, array or matrix
min	lower bound
max	upper bound
direction	string indicating from real to transformed "RT" or vice-versa "TR". For direction "TR" the lower and upper bounds are required inputs.

Value

transformed or re-transformed data of same format as x

Note

todo: function(x,min = min(x),max = max(x),direction = "RT") does not work, other solutions? default values for min and max only valid if direction is also RT, how can this be required

Examples

```
vec <- runif(10, -5, 10)
norm_minmax(vec, min(vec), max(vec))
vecT <- norm_minmax(vec, -5, 10)
# re-scale to vec
norm_minmax(vecT, -5, 10, "TR")
```

optim_fn

Function to be optimized during calibration

Description

Function that takes global parameters (see Details) and performs optimization to find best parameter set

Usage

```
optim_fn(
  ParamOptim,
  hydro_data,
  split_indices,
  model,
  input,
  error_crit,
  cal_maximize,
  cal_q_transfo,
  do_transfo_param
)
```

Arguments

ParamOptim	vector of model parameters to be optimized
hydro_data	a list or data frame containing observed runoff loaded with load_meteo_data
split_indices	indices indicating the warm-up and calibration period, i.e. the output of split_data_set
model	a string indicating the hydrological model
input	model input returned by create_input
error_crit	a string of a function indicating an error criterion, e.g. KGE , or others from the hydroGOF which are usable by calc_hydroGOF functions
cal_maximize	a logical indicating if function should be maximized
cal_q_transfo	a string indicating how runoff should be transformed with transfo_q
do_transfo_param	logical indicating whether calibration is done with real or transformed parameters

Details

Simulates a hydrological model during warm-up and calibration period, and calculates an error criterion `error_crit` on the calibration period only.

Replace NA in error criterion, which is generated when simulated runoff is

- solely NAs: set a particularly bad value, i.e. +/- 1e10 (dependent on minimization or maximization)
- solely 0s for [KGE](#): set to asymptotic value, i.e. $1 - \sqrt{3}$
- for all other cases it throws an error

Value

error criterion to be optimized

Note

: Qsim is converted to numeric to similar to Qobs, this is important for TUW for example, which returns a matrix 1 x time, for the lumped case. todo:

1. include a spatial explicit version
2. exclude warning for other NA cases in error criterion as described above
3. Consider to link KGE etc with `cal_maximize`

See Also

[calibrate_model](#), [calibrate_model](#), [create_input](#), [load_meteo_data](#), [calc_hydroGOF](#)

Examples

```
# differential evolution
cal_results <- DEoptim::DEoptim(
  fn = optim_fn, hydro_data = hydro_data, split_indices = split_indices,
  model = model, input = input, error_crit = error_crit, cal_maximize = cal_maximize,
  cal_q_transfo = cal_q_transfo, do_transfo_param = do_transfo_param,
  lower = lower, upper = upper,
```

```

control = DEoptim::DEoptim.control(
  NP = cal_par[[model]][["DEoptim"]][["NP"]],
  itemax = cal_par[[model]][["DEoptim"]][["itemax"]],
  trace = 10
)

```

save_airGR_plot

save an airGR overview plot to pdf

Description

use the [airGR plot](#) function

Usage

```
save_airGR_plot(file, simulation_results, Qobs)
```

Arguments

file	the filename of the saved plot
simulation_results	output from simulate_model , i.e. the output object of the airGR function RunModel
Qobs	a vector of observed runoff

Value

a logical if file was written successfully

Examples

```
save_airGR_plot("airGR_plot.pdf", simulation_results, Qobs)
```

save_cal_val_plot

Save a calibration validation plot

Description

Create and save a plot containing over the calibration and validation period

Usage

```
save_cal_val_plot(file, BasinObs, Qsim, split_indices)
```

Arguments

file	the filename of the saved plot
BasinObs	data frame with time series of input data from load_meteo_data
Qsim	vector with simulated runoff
split_indices	a list of indices from split_data_set containing the elements ind_cal and ind_val

Details

Plots observed and simulated runoff and displays certain error metrics

Value

a logical if the file was successfully written

Note

todo: include a seasonal rolling mean

Examples

```
save_cal_val_plot("cal_val.pdf", BasinObs, split_indices)
```

set_airGR_par

Create default calibration settings for the airGR family

Description

Create default calibration settings for the airGR family

Usage

```
set_airGR_par(model)
```

Arguments

model a model string

Value

a list of calibration settings

Examples

```
set_airGR_par("GR4J")
```

simulate_model	<i>Simulate a hydrological model</i>
----------------	--------------------------------------

Description

Takes a model choice, model parameters, model input and indices indicating a time subset of the model input which should be used and simulates discharge values and model specific output

Usage

```
simulate_model(model, model_param, input, ind = seq_along(input[[1]]))
```

Arguments

model	a string specifying the hydrological model, currently implemented are airGR and TUWmodel package models
model_param	a vector of model parameters specific for each model choice
input	the output of the create_input function, dependent on the model choice, in general containing information about date, precipitation air temperature and potential evapotranspiration and spatial information as area of the catchment or hypsometric curves
ind	a vector of indices which can be used to subset the time dependent elements in input. The default is taking all indices from the first element of input

Value

a list with the elements Qsim, the simulated runoff, and more_info with model specific output

Examples

```
simulation_results <- simulate_model("TUWmodel", calibration_results$model_param, input, ind = split_indices$
```

split_data_set	<i>split a date vector</i>
----------------	----------------------------

Description

Splits a date vector hardcoded into three time periods, i.e. warm up, calibration and validation period

Usage

```
split_data_set(date, start_end_date_vec)
```

Arguments

date	a vector of dates (e.g. Dates, Posix)
start_end_date_vec	vector of length six, with entries start and end date for warm-up, start and end date for calibration and start and end date for validation, in this order.

Value

a list with three vectors containing indices ind_warm, ind_cal, ind_val

Examples

```
split_data_set(
  seq(from = as.Date("1981-01-01"), to = as.Date("2020-12-31"), by = "days"),
  c("1981-01-01", "1982-12-31", "1983-01-01", "2000-12-31", "2001-01-01", "2020-12-31")
)
```

transfo_param	<i>Parameter transformation</i>
---------------	---------------------------------

Description

Transforms model parameters to hypercube and vice versa

Usage

```
transfo_param(param, direction, model, cal_parameter = default_cal_par)
```

Arguments

- param vector of model parameters
- direction string indicating from real to transformed "RT" or vice-versa "TR".
- model a string specifying the hydrological model

Details

For airGR models the functions [TransfoParam](#) of package [airGR](#) are applied. Model combinations as "CemaNeigeGR4J" are allowed.

For other models the parameter space is transformed to [0,1] with [norm_minmax](#)

Value

a vector with transformed parameters

Note

todo: 1) CemaNeigeHyst is not implemented yet 2) how do the links work without airGR installed?
3) implement functionality that airGR functions are not used at all

See Also

[TransfoParam](#)

Examples

```
# scale a parameter set for model "TUV" to [0,1] and back
param <- c(1, 2, 3, -1, 1, 1, 200, 10, 1, 15, 100, 50, 2, 15, 50)
scaled <- transfo_param(param, "RT", "TUV")
rescaled <- transfo_param(scaled, "TR", "TUV")
# scale a parameter set for "CemaNeigeGR4J" to [-9.99,9.99] and back
param <- c(1000, 2, 250, 7, .2, 109.0365)
scaled <- transfo_param(param, "RT", "CemaNeigeGR4J")
rescaled <- transfo_param(scaled, "TR", "CemaNeigeGR4J")
```

transfo_q	<i>Transforms runoff</i>
-----------	--------------------------

Description

Inverse, sqrt and no transform of runoff, log is not advised for KGE, see airGR or Santos 2018

Usage

```
transfo_q(Q, q_transfo_type = "none")
```

Arguments

Q vector, matrix, data.frame etc of runoff values

q_transfo_type a string indicating how the runoff should be transformed. Currently "none", "sqrt", "inv" and "log" are supported.

Value

transformed runoff in same format as input

Note

todo: discuss what to do with Infinity

Examples

```
transfo_q(array(0:10, c(2, 5)), "log")
```

validate_model	<i>Validate model</i>
----------------	-----------------------

Description

Calculates validation measures for different transformation types

Usage

```
validate_model(Qsim, Qobs, val_crit_transfo = "KGE_none")
```

Arguments

Qsim	vector with simulated runoff
Qobs	vector with observed runoff
val_crit_transfo	a vector of strings specifying validation criteria and a runoff transformation separated by a "_". Supported are validation criteria from the hydroGOF package usable by the calc_hydroGOF function, for supported runoff transformations please refer to transfo_q

Value

a long data frame with columns crit indicating the used validation criterion, transfo for the used runoff transformation and value.

Examples

```
validate_model(
  1:10, seq(2, 11),
  c("KGE_log", "NSE_inv", "VE_none", "pbias_none")
)
```

write_ascii	<i>Writes ascii results</i>
-------------	-----------------------------

Description

Writes an ascii overview of parameters and validation results

Usage

```
write_ascii(
  file,
  calibration_results,
  validation_results,
  equally_spaced = TRUE
)
```

Arguments

`file` filename
`calibration_results` a list containing the calibration results from [calibrate_model](#). Only a vector with calibrated model parameters are written out
`validation_results` a data frame containing the validation results from [validate_model](#).
`equally_spaced` (optional) a logical indicating if a equally spaced output should be written out

Value

a logical if the file was successfully written

Examples

```
write_ascii("results.txt", calibration_results, validation_results)
```

Index

* datasets

default_cal_par, 8
hydro_family, 9

airGR, 5, 13, 16

calc_hydroGOF, 2, 3–5, 7, 12, 18
calc_subseasonal_validation_results, 2
calc_validation_results, 2, 3, 4
calibrate_model, 5, 12, 19
Calibration_Michel, 5
call_cal_fn, 6, 6
create_input, 7, 7, 12

default_cal_par, 8
DEoptim, 5

find_monthly_indices, 8

get_family, 9

hydro_family, 9
hydroGOF, 2–5, 7, 12, 18
hydroPSO, 5

KGE, 7, 12

load_meteo_data, 3, 5, 7, 10, 12, 13

malschains, 5

norm_minmax, 10, 16

optim_fn, 6, 11

plot, 13

RunModel, 13

save_airGR_plot, 13
save_cal_val_plot, 13
set_airGR_par, 14
simulate_model, 13, 15
split_data_set, 5, 7, 12, 13, 15

transfo_param, 16
transfo_q, 3–5, 7, 12, 17, 18

TransfoParam, 16

validate_model, 3, 4, 18, 19

write_ascii, 18