Package 'dycdtools'

May 8, 2026

Title Calibration Assistant and Post-Processing Tool for Aquatic Ecosystem Model DYRESM-CAEDYM

Version 0.4.4

Description Dynamic Reservoir Simulation Model (DYRESM) and Computational Aquatic Ecosystem Dynamics Model (CAEDYM) model development, including assisting with calibrating selected model parameters and visualising model output through time series plot, profile plot, contour plot, and scatter plot. For more details, see Yu et al. (2023) <https://journal.r-project.org/articles/RJ-2023-008/>.

URL https://github.com/SongyanYu/dycdtools

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr, ncdf4, tidyr, ggplot2, RColorBrewer, lubridate, R.utils, parallel

RoxygenNote 7.2.2

Depends R (>= 2.10)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Songyan Yu [aut, cre] (ORCID: <https://orcid.org/0000-0001-5765-7060>), Christopher McBride [ctb], Marieke Frassl [ctb]

Maintainer Songyan Yu <yusongyan1989@gmail.com>

Repository CRAN

Date/Publication 2024-09-11 14:50:02 UTC

Contents

`calib_assist` *Assist calibration of DYRESM-CAEDYM model.*

Description

This function carries out simulations with a large number of possible combinations of parameter values that users regard as potentially suitable for their model calibration, and calculates the values of nominated objective functions (i.e., statistical measures of goodness of fit) for each combination. Based on the calculated objective function values, users can determine the optimal set(s) of parameter values or narrow the ranges of possible parameter values.

Usage

```
calib_assist(
  cal.para,
  combination = "random",
  n,
  model.var,
  phyto.group = NA,
  obs.data,
  objective.function = c("NSE", "RMSE"),
  start.date,
  end.date,
  dycd.wd,
  dycd.output,
  file.name,
  verbose = TRUE,
  parallel = FALSE,
  n.cores = NULL,
  write.out = TRUE
)
```

Arguments

cal.para	a data frame or a character string naming an external .csv file where below column names are mandatory: "Parameter" describing parameter names (abbreviation is allowed), "Min", "Max", and "Increment" describing the minimum and maximum parameter values and expected increment in the value range, "Input_file" and "Line_NO" listing in which configuration file at which line the parameter can be found.
combination	a vector of string character of how to pick up combinations of parameter values. "random" - the function randomly picks up a given number of combinations; "all" - the function tries all possible combinations of parameter values.
n	the number of random selections. Must be provided if combination = "random".
model.var	a vector of string character of modelled variables for calibration. the character should be from the 'var.name' column of 'data(output_name)'. Note that if model calibration needs to regard chlorophyll of multiple phytoplankton groups as a whole, model.var should use "CHLA" and individual phytoplankton group should be specified through the "phyto.group" argument. If phytoplankton groups are separately calibrated, simply list their character in this argument (model.var).
phyto.group	a vector of simulated phytoplankton groups, including CHLOR, FDIAT, NODUL, CYANO and CRYPT.
obs.data	a data frame or a character string naming a csv file of observed lake data. The observed lake data need to include below columns: 1) 'Date' in format of "%Y-%m-%d" 2) 'Depth' (integer) 3) Water quality variables (use string characters of model var as column names). see example data 'data(obs_temp)'.
objective.function	a vector of string character describing which objective function(s) to be used for calibration. Selected from the following five functions: "NSE": Nash-Sutcliffe efficiency coefficient, "RMSE": Root Mean Square Error, "MAE": Mean Absolute Error, "RAE": Relative Absolute Error, "Pearson": Pearson's r.
start.date, end.date	the beginning and end simulation dates for the intended DYRESM-CAEDYM calibration. The date format must be "%Y-%m-%d". The two dates should be consistent with model configurations.
dycd.wd	the directory where input files (including the bat file) to DYRESM-CAEDYM are stored.
dycd.output	a character string naming the output file of model simulation.
file.name	a character string naming a .csv file where the results of this function are written to. Needed if 'write.out' = TRUE.
verbose	if TRUE, model calibration information is printed.
parallel	if TRUE, the calibration process is run on multiple cores.
n.cores	When 'parallel' is TRUE, n.cores is the number of cores the calibration function will be run on. If not provided, the default value is the number of available cores on the computer -1.
write.out	if TRUE, model calibration results are saved in a file with a file name set by the "file.name" argument.

Value

a data frame of all tested values of parameters and corresponding values of the objective function(s).

Note

No executable examples are provided to illustrate the use of this function, as this function relies on the DYRESM-CAEDYM executables to work.

`change_input_file` *Change parameter value of input files to DYRESM_CAEDYM model.*

Description

Change parameter value of input files to DYRESM_CAEDYM model.

Usage

```
change_input_file(input_file, row_no, new_value)
```

Arguments

`input_file`	vector of input format, such as "par","cfg".
`row_no`	the number of row where the variable of interest is in the input file.
`new_value`	the new value that will be assigned to the variable of interest.

Value

updated input_file with a new value to a parameter.

`delete_space` *Delete all whitespace until a non-whitespace character.*

Description

Delete all whitespace until a non-whitespace character.

Usage

```
delete_space(extract_val)
```

Arguments

`extract_val`	a vector.

ext_output *Extract outputs from a DYRESM-CAEDYM model run*

Description

Extract simulation outputs from a DYRESM-CAEDYM model run.

Usage

```
ext_output(dycd.output, var.extract, verbose = FALSE)
```

Arguments

dycd.output	a string of characters describing the file path to the output netcdf file of DYRESM-CAEDYM model.
var.extract	a vector of variables to be extracted from the output. Please refer to the var.name of data(output_name) for accepted variable name. Apart from the user nominated variables, simulation period and layer height data are also extracted.
verbose	if TRUE, the information about the extraction process is printed.

Value

a list of values of those variables of interest, as well as two compulsory variables (i.e. simulation period, layer height)

Examples

```
# extract simulated temperature values from DYRESM-CAEDYM simulation file
var.values<-ext_output(dycd.output=system.file("extdata", "dysim.nc",
                                               package = "dycdtools"),
                       var.extract=c("TEMP"))

for(i in 1:length(var.values)){
 expres<-paste0(names(var.values)[i],"<-data.frame(var.values[[",i,"]])")
 eval(parse(text=expres))
}
```

`hgt_to_dpt` *convert from height to depth*

Description

convert from height to depth

Usage

```
hgt_to_dpt(height)
```

Arguments

`height`	a vector of height profile

`interpol` *Interpolation of DYRESM-CAEDYM simulation results across a series of user-defined depths.*

Description

The default simulation results of a water quality variable from DYRESM-CAEDYM are usually at irregular layer heights. This function convert it to a data frame with regular layer heights through interpolation.

Usage

```
interpol(layerHeights, var, min.depth, max.depth, by.value)
```

Arguments

`layerHeights`	layer heights, outputs from a DYRESM-CAEDYM model run, and can be generated with the 'ext_output' function.
`var`	simulation results of a water quality variable and can also be generated with the 'ext_output' function.
`min.depth, max.depth, by.value`	minimum and maximum layer depths within which interpolation will be conducted. by.value sets up the depth increments between two immediate layers.

Value

a matrix of interpolated values of the water quality variable(s).

Examples

```
# extract simulated temperature values from DYRESM-CAEDYM simulation file
 var.values<-ext_output(dycd.output=system.file("extdata", "dysim.nc",
                                                package = "dycdtools"),
                        var.extract=c("TEMP"))

 for(i in seq_along(var.values)){
  expres<-paste0(names(var.values)[i],"<-data.frame(var.values[[",i,"]])")
  eval(parse(text=expres))
 }

 # interpolate temperature for depths from 0 to 13 m at increment of 0.5 m
 temp.interpolated<-interpol(layerHeights = dyresmLAYER_HTS_Var,
                             var = dyresmTEMPTURE_Var,
                             min.dept = 0,
                             max.dept = 13,
                             by.value = 0.5)
```

`objective_fun` *Calculate statistical measures of goodness of fit for DYRESM-CAEDYM model simulations.*

Description

calculate the below five objective functions that are commonly used to measure goodness of fit: 1) Nash-Sutcliffe Efficiency coefficient (NSE), 2) Root Mean Square Error (RMSE), 3) Mean Absolute Error (MAE), 4) Relative Absolute Error (RAE), and 5) Pearson's r (Pearson).

Usage

```
objective_fun(
  sim,
  obs,
  fun = c("NSE", "RMSE"),
  start.date,
  end.date,
  min.depth,
  max.depth,
  by.value
)
```

Arguments

`sim`	a matrix of a simulated water quality variable values with column of time and row of depth. This matrix can be generated by running the "interpol" function.

obs	a data frame having three columns to describe observed values of a water quality variable. These three columns are 'Date' (as '%Y-%m-%d'), 'Depth', and the designated variable name which can be found from the var.name column of 'data(output_name)'. An example of such a data frame can be found with 'data(obs_temp)'
fun	objective function(s) to be calculated. Select any from 'NSE', 'RMSE', 'MAE', 'RAE', and 'Pearson'. Multiple selections are allowed.
start.date, end.date	the start and end simulation dates for the DYRESM-CAEDYM model run. The date format must be "%Y-%m-%d".
min.depth, max.depth	the minimum and maximum depths of the simulation matrix.
by.value	the value of increment at which the depth of layers increases from the mim.depth to max.depth in the simulation matrix.

Value

a list of objective function values.

---

obs_temp *Example observed profiling temperature data across different depths over the period of 6-11 June 2017.*

Description

A table has three columns. The first column name is Date in the form of dd-mm-YY. The second column is Depth where the temperature data was monitored. The third column is monitored temperature value.

Usage

```
data(obs_temp)
```

Format

A data frame with 77 rows and 3 variables:

Date date when the monitoring happened

Depth depth of monitoring

TEMP temperature value

Source

self-made.

output_name *Default DYCD simulation variable names with their variable name*

Description

A table has two columns. The first column name is var.name, meaning variable names that are used in the extract.output function. The second column is the default DYCD simulation variable names, such as "dyresmLAYER_HTS_Var".

Usage

```
data(output_name)
```

Format

A data frame with 65 rows and 2 variables:

var.name variable name

output.name default DYCD simulation variable name

Source

self-made.

plot_cont *Contour plot of only simulation results of a water quality variable.*

Description

Contour plot a matrix of values of a water quality variable,

Usage

```
plot_cont(
  sim,
  sim.start,
  sim.end,
  legend.title,
  min.depth,
  max.depth,
  by.value,
  nlevels
)
```

Arguments

`sim`	a matrix of simulated variables. This matrix can be generated by running the "interpol" function.
`sim.start`, `sim.end`	the start and end dates of the simulation period for the DYRESM-CAEDYM model run of interest. The date format must be "%Y-%m-%d".
`legend.title`	the legend title of the contour figure.
`min.depth`, `max.depth`, `by.value`	minimum and maximum depths used to be the start of y axis of the contour plot, at the increment of by.value.
`nlevels`	Number of levels which are used to partition the range of simulation variable.

Details

This function is NOT based on ggplot2. To save the produced figure, users can use functions like png, bmp, jpeg, etc.

Value

This function returns a filled.contour object.

Examples

```
sim <- matrix(c(28,28,28,27,25,24,12,13,14,15,16,17),
              nrow = 6,
              ncol = 2)

# contour plot of the sim data frame
  p <- plot_cont(sim = sim,
            sim.start = "2020-01-01",
            sim.end = "2020-01-02",
            legend.title = "T \u00B0C",
            min.depth = 0, max.depth = 5, by.value = 1,
            nlevels = 20)

  p
```

---

plot_cont_comp *Contour plot of DYRESM-CAEDYM simulation outputs of a water quality variable, with observed data shown as dots in the generated contour plot.*

Description

Contour plot a matrix of values of a water quality variable.

Usage

```
plot_cont_comp(
  sim,
  obs,
  sim.start,
  sim.end,
  plot.start,
  plot.end,
  legend.title,
  min.depth,
  max.depth,
  by.value,
  nlevels = 20
)
```

Arguments

sim	a matrix of simulated variables. This matrix can be generated by running the "interpol" function.
obs	a data frame having three columns to describe observed values of a water quality variable. These three columns are 'Date' (as '%Y-%m-%d'), 'Depth', and the designated variable name which can be found from the var.name column of 'data(output_name)'. An example of such a data frame can be found with 'data(obs_temp)'
sim.start, sim.end	the start and end dates of the simulation period for the DYRESM-CAEDYM model run of interest. The date format must be "%Y-%m-%d".
plot.start, plot.end	the start and end dates of the period to be plotted, in the format of "%Y-%m-%d".
legend.title	the legend title of the contour figure.
min.depth, max.depth, by.value	minimum and maximum depths used to be the start of y axis of the contour plot, at the increment of by.value.
nlevels	Number of levels which are used to partition the range of simulation variable.

Details

This function is NOT based on ggplot2. To save the produced figure, users can use functions like png, bmp, jpeg, etc.

Value

This function returns a filled.contour object.

Examples

```
obs <- data.frame(Date = c(rep('2020-01-01', 6), rep('2020-01-02', 6)),
                  Depth = rep(0:5, 2),
```

```
                  TEMP = rep(29:24,2))

sim <- matrix(c(28,28,28,27,25,24,12,13,14,15,16,17),
              nrow = 6,
              ncol = 2)

# contour plot of temperature simulations
# with observed data shown as colour-coded dots
  p <- plot_cont_comp(sim = sim,
                 obs = obs,
                 sim.start = "2020-01-01",
                 sim.end = "2020-01-02",
                 plot.start = "2020-01-01",
                 plot.end = "2020-01-02",
                 legend.title = "T \u00B0C",
                 min.depth=0, max.depth=5, by.value=1,
                 nlevels=20)

 p
```

---

plot_prof *A post-processing function used to visualise model output in a profile graph.*

Description

Profile plot shows vertical profiles of simulation outputs and corresponding observations for all dates where observations are available.

Usage

```
plot_prof(
  sim,
  obs,
  sim.start,
  sim.end,
  plot.start,
  plot.end,
  xlabel,
  min.depth,
  max.depth,
  by.value
)
```

Arguments

sim	a matrix of simulated variables. This matrix can be generated by running the "interpol" function.

obs
: a data frame having three columns to describe observed values of a water quality variable. These three columns are 'Date' (as '%Y-%m-%d'), 'Depth', and the designated variable name which can be found from the var.name column of 'data(output_name)'. An example of such a data frame can be found with 'data(obs_temp)' This function is based on ggplot2, and users can treat the object of this function in the same way as a ggplot2 object.

sim.start, sim.end
: the start and end dates of the simulation period of the DYRESM-CAEDYM model run of interest. The date format must be "%Y-%m-%d".

plot.start, plot.end
: the start and end dates of the period to be plotted in the format of "%Y-%m-%d".

xlabel
: the x axis label of the profile figure.

min.depth, max.depth, by.value
: minimum and maximum depths in the profile plot at an increment of by.value.

Value

This function returns a ggplot object that can be modified with ggplot package functions.

Examples

```
 var.values<-ext_output(dycd.output=system.file("extdata", "dysim.nc",
                                                package = "dycdtools"),
                        var.extract=c("TEMP"))

 for(i in 1:length(var.values)){
   expres<-paste0(names(var.values)[i],"<-data.frame(var.values[[",i,"]])")
   eval(parse(text=expres))
  }

# interpolate temperature for depths from 0 to 13 m at increment of 0.5 m
  temp.interpolated<-interpol(layerHeights = dyresmLAYER_HTS_Var,
                              var = dyresmTEMPTURE_Var,
                              min.dept = 0, max.dept = 13, by.value = 0.5)

 data(obs_temp)
# profile plot of temperature sim and obs
  p <- plot_prof(sim=temp.interpolated,
            obs = obs_temp,
            sim.start="2017-06-06",
            sim.end="2017-06-15",
            plot.start="2017-06-06",
            plot.end="2017-06-15",
            xlabel = "Temperature \u00B0C",
            min.depth = 0, max.depth = 13, by.value = 0.5)
 p
```

plot_scatter *Scatter plot of the simulation and observation of a water quality variable. This function is based on ggplot2, and users can treat the object of this function in the same way as a ggplot2 object.*

Description

Scatter plot of the simulation and observation of a water quality variable. This function is based on ggplot2, and users can treat the object of this function in the same way as a ggplot2 object.

Usage

```
plot_scatter(
  sim,
  obs,
  sim.start,
  sim.end,
  plot.start,
  plot.end,
  min.depth,
  max.depth,
  by.value
)
```

Arguments

sim	a matrix of simulated variables. This matrix can be generated by running the "interpol" function.
obs	a data frame having three columns to describe observed values of a water quality variable. These three columns are 'Date' (as '%Y-%m-%d'), 'Depth', and the designated variable name which can be found from the var.name column of 'data(output_name)'. An example of such a data frame can be found with 'data(obs_temp)'
sim.start, sim.end	the start and end dates of the simulation period of the DYRESM-CAEDYM model run of interest. The date format must be "%Y-%m-%d".
plot.start, plot.end	the start and end dates of the period to be plotted in the format of "%Y-%m-%d".
min.depth, max.depth, by.value	minimum and maximum depths in the profile plot at an increment of by.value.

Value

This function returns a ggplot object that can be modified with ggplot package functions.

Examples

```
var.values<-ext_output(dycd.output=system.file("extdata", "dysim.nc",
                                               package = "dycdtools"),
                       var.extract=c("TEMP"))

for(i in 1:length(var.values)){
  expres<-paste0(names(var.values)[i],"<-data.frame(var.values[[",i,"]])")
  eval(parse(text=expres))
 }

# interpolate temperature for depths from 0 to 13 m at increment of 0.5 m
  temp.interpolated<-interpol(layerHeights = dyresmLAYER_HTS_Var,
                              var = dyresmTEMPTURE_Var,
                              min.dept = 0, max.dept = 13, by.value = 0.5)

data(obs_temp)

# scatter plot of sim and obs temperature
p <- plot_scatter(sim=temp.interpolated,
             obs=obs_temp,
             sim.start="2017-06-06",
             sim.end="2017-06-15",
             plot.start="2017-06-06",
             plot.end="2017-06-15",
             min.depth = 0, max.depth = 13, by.value = 0.5)

p
```

`plot_ts` *Time series plot of simulated and observed values at target depths. This function is based on ggplot2, and users can treat the object of this function in the same way as a ggplot2 object.*

Description

Time series plot of simulated and observed values at target depths. This function is based on ggplot2, and users can treat the object of this function in the same way as a ggplot2 object.

Usage

```
plot_ts(
  sim,
  obs,
  target.depth,
  sim.start,
  sim.end,
  plot.start,
  plot.end,
```

```
  min.depth,
  max.depth,
  by.value,
  ylabel
)
```

Arguments

sim	a matrix of simulated variables. This matrix can be generated by running the "interpol" function.
obs	a data frame having three columns to describe observed values of a water quality variable. These three columns are 'Date' (as '%Y-%m-%d'), 'Depth', and the designated variable name which can be found from the var.name column of 'data(output_name)'. An example of such a data frame can be found with 'data(obs_temp)'
target.depth	a vector of depth (unit:m) for which time series simulation results will be plotted.
sim.start, sim.end	the start and end dates of the simulation period of the DYRESM-CAEDYM model run of interest. The date format must be "%Y-%m-%d".
plot.start, plot.end	the start and end dates of the period to be plotted in the format of "%Y-%m-%d".
min.depth, max.depth, by.value	minimum and maximum depths in the profile plot at an increment of by.value.
ylabel	the y axis title of the time series plot.

Value

This function returns a ggplot object that can be modified with ggplot package functions.

Examples

```
 var.values<-ext_output(dycd.output=system.file("extdata", "dysim.nc",
                                                package = "dycdtools"),
                        var.extract=c("TEMP"))

 for(i in 1:length(var.values)){
   expres<-paste0(names(var.values)[i],"<-data.frame(var.values[[",i,"]])")
   eval(parse(text=expres))
  }

# interpolate temperature for depths from 0 to 13 m at increment of 0.5 m
  temp.interpolated<-interpol(layerHeights = dyresmLAYER_HTS_Var,
                              var = dyresmTEMPTURE_Var,
                              min.dept = 0, max.dept = 13, by.value = 0.5)

 data(obs_temp)
# time series plot of temperature sim and obs
 p <- plot_ts(sim = temp.interpolated,
              obs = obs_temp,
```

```
                target.depth=c(1,6),
                sim.start="2017-06-06",
                sim.end="2017-06-15",
                plot.start="2017-06-06",
                plot.end="2017-06-15",
                ylabel="Temperature \u00B0C",
                min.depth=0,
                max.depth=13,
                by.value=0.5)

    p
```

`run_iteration` *Internal function to provide parallel processing support to the calibration assistant function.*

Description

Internal function to provide parallel processing support to the calibration assistant function.

Usage

```
run_iteration(this.sim, dycd.wd)
```

Arguments

`this.sim`	a numeric denoting which parameter combination to be tried.
`dycd.wd`	working directory where input files (including the bat file) to DYRESM-CAEDYM are stored.

Index