Package 'CQ2'

September 17, 2025

Type Package

Title Objective Calibration of Quick-Slow CQ Models and Baseflow

Version 0.1.0

Author Thomas Westfall, ORCID = ``0009-0000-0529-881X"

Maintainer Thomas Westfall <thomas.westfall1@monash.edu>

Description The CQ2 package evaluates a suite of C-Q models with single and multiple flow components on daily observations of streamflow and concentration. Models are globally calibrated with maximum likelihood estimation providing an objective estimate of baseflow in models with a slow-flow component. Calibrated models are compared to eachother and alongside observations with various plots and metrics (AIC, NSE, RMSE). There are 15 C-Q models available (C1 - C15). The default analysis will compare C1, the simple C-Q model, and C13, a quick-slow version of the Hubbard Brook working model, that was the best performing at explaining the varability in the C-Q relationship. Derivations and explanations of each model are located in the following article: Westfall T.G., Peterson T.J., Lintern A., Western, A.W (2024), Slow and quick flow models explain the temporal dynamics of daily salinity in streams https://doi.org/10.1029/2024WR039103.

Imports cmaesr, padr, Hmisc, checkmate

URL https://github.com/ThomasWestfall/CQ2

License GPLv3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Depends R (>= 3.5)

R topics documented:

CQ2-package	*Overview of methods and procedures*

Description

CQ2 fits and compares C-Q models with single flow components and slow-quick flow components on daily streamflow and concentration observations. The slow flow component in the multiple flow component C-Q models is estimated as baseflow using the Eckhardt (2005) baseflow filter, and the filter parameters are objectivly calibrated along with model parameters using global optimization with the cmaesr (R-package). You have the option to run and evaluate any or all of the 15 models that are provided (Chat1 - Chat15), but the default analysis will compare Chat1, the simple C-Q model $C = aQ^b$ and Chat13, a slow-quick version of the Hubbard Brook working model, that was the best performing at explaining the varation in C-Q plots. See the article when published: Westfall T.G., Peterson T.J., Lintern A., Western, A.W (2024), Slow and quick flow models explain the temporal dynamics of daily salinity in streams (IN PREP)

The CQ2 package operates by first setting up the data and models, `setModels`, then fitting the models, `runModels`. Be prepared to run models over lunch or overnight as the computation time for the slow-quick models can take several hours for 20+ year records of daily data. After computation, output the predictions from the models into a dataframe with `getResults`. Then the results can be evaluated through comparing the statistics (i.e. AIC, NSE, RMSE, BFI), `getStats`, parameters, `getParam`, or plot the predictions in C-Q scatter plots and annual timeseries along with observations and the objectivly estimated baseflow, `plotResults`!

I. Set-Up

setModels	set-up data and C-Q models

II. Fit

runModels	fit C-Q models

III. Review

getResults	retrieve predictions from C-Q models
plotResults	plot predictions from C-Q models
getStats	retrieve performance of each C-Q model
getParam	retrieve parameters from C-Q models

Authors

Except where indicated otherwise, the methods and functions in this package were written by Thomas Westfall.

Acknowledgments

Immense graditude to Tim Peterson, Anna Lintern, Andrew W. Western, and Lucas Pamminger for their assistance, patience, and support.

`getParam` *Get Param*

Description

getParam outputs the parameters for each fitted C-Q model

Usage

```
getParam(
  model.setup = list(),
  Chat.model.name = character(),
  output.data = dataframe()
)
```

Arguments

`model.setup`	lists of details about data, model, and site from `setModels`
`Chat.model.name`	character string of one fitted model (i.e. "C13")
`output.data`	dataframe of daily concentration and baseflow predictions from `getResults`

Details

`getParam`

exported datafame with parameters from fitted C-Q models

Value

output summary dataframe with parameters for fitted models

getResults *Get Results*

Description

getResults of fitted C-Q models

Usage

```
getResults(model.setup = list())
```

Arguments

`model.setup`	lists of details about data, model, and site from setModels()
`cmaes.results`	list of cmaes.results from fitted models

Details

`getResults`

exported predicted concentration and baseflow of fitted models

Value

output original data_all dataframe with predicted concentration and baseflow of fitted models

getStats *Get Stats*

Description

getStats calculates the statistics for each fitted C-Q model

Usage

```
getStats(model.setup = list(), output.data = dataframe())
```

Arguments

`model.setup`	lists of details about data, model, and site from `setModels`
`output.data`	dataframe of daily concentration and baseflow predictions from `getResults`

Details

`getStats`

exported summary table with negLL, AIC, NSE, RMSE, and BFI

Value

output summary dataframe with statistics (negLL, AIC, NSE, RMSE, and BFI)

plotResults *plot Results*

Description

plotResults plots predictions from each fitted C-Q model

Usage

```
plotResults(
  model.setup = list(),
  output.data = data.frame(),
  plot.models = character(c()),
  plot.type = "scatter"
)
```

Arguments

`model.setup`	lists of details about data, model, and site from setModels()
`output.data`	dataframe of daily concentration and baseflow predictions from `getResults`
`plot.models`	character string vector with a 'C#' model name from provided models (i.e. C1-C15). Ch1 and C13 default. Two model limit.
`plot.type`	character string of either "scatter" or "timeseries" to view results. Note, 'timeseries' exports plot as a pdf in the working directory

Details

`plotResults`

plots C-Q scatter plots and annual timeseries with predictions from fitted C-Q models

Value

annual timeseries plot comparing predictions from two models with observations, streamflow and baseflow; C-Q scatter plots of each model

runModels *Run models*

Description

Run C-Q models for comparison

Usage

```
runModels(model.setup = list())
```

Arguments

`model.setup`	lists of details about data, model, and site from setModels()

Details

`runModels`

Runs C-Q models from the selection of Chat1-15. Default runs simple C-Q model (Chat1) and quick-slow Hubbard Brook model (Chat13)

Value

fitted C-Q models

`setModels` *Set models*

Description

Set-up C-Q models for comparison

Usage

```
setModels(
  Chat.model.names = character(),
  input.data = data.frame(year = c(), month = c(), day = c(), C = c(), Q = c()),
  Qthresh = 0,
  Likelihood.name = "GaussLiklihood",
  site.id = character(),
  site.name = character()
)
```

Arguments

`Chat.model.names` character string vector with a 'Chat#' model name from provided models (i.e. Chat1-Chat15). Chat1 and Chat13 default

`input.data` dataframe of daily runoff and concentration. colnames = c("year", "month", "day", "C", "Q")

`Qthresh` numeric low-flow streamflow threshold, models only fitted to observations with same day streamflow above this threshold.

`Likelihood.name` character string with name of likelihood function ("GaussLiklihood","GaussLiklihoodAR1", or "GaussLiklihoodAR3")

`site.id` character string with identifier of gauge or catchment

`site.name` character string with name of gauge or catchment

Details

`setModels`

Sets-up C-Q models from the selection of Chat1-15. Default sets-up simple C-Q model (Chat1) and quick-slow Hubbard Brook model (Chat13)

Value

C-Q models ready for runModels

Index