

Package ‘CQ2’

August 23, 2024

Type Package

Title Objective Calibration of Baseflow and Quick-Slow CQ Models

Version 0.1.0

Author ``Thomas Westfall"

Maintainer The package maintainer <yourself@somewhere.net>

Description More about what it does (maybe more than one line)

Use four spaces when indenting paragraphs within the Description.

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

R topics documented:

CQ2-package	1
getParam	3
getResults	3
getStats	4
plotResults	5
runModels	5
setModels	6

Index	7
--------------	----------

CQ2-package	<i>Overview of methods and procedures</i>
-------------	---

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 objectively 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 variation in C-Q plots. See the article when published: Westfall T., Peterson TJ., Lintern A., Western, A.W (2024), Slow and quick flow models explain the temporal dynamics of daily salinity in streams (IN REVIEW)

The CQ2 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 objectively estimated baseflow, `plotResults`!

I. Set-Up

<code>setModels</code>	set-up data and C-Q models
------------------------	----------------------------

II. Fit

<code>runModels</code>	fit C-Q models
------------------------	----------------

III. Review

<code>getResults</code>	retrieve predictions from C-Q models
<code>plotResults</code>	plot predictions from C-Q models
<code>getStats</code>	retrieve performance of each C-Q model
<code>getParam</code>	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 gratitude to Tim Peterson, Anna Lintern, Andrew W. Western, and Lucas Pamminger for assistance and support.

`getParam`*Get Param*

Description

getParam outputs the parameters for each fitted C-Q model

Usage

```
getParam(  
  Chat.model.names = c("Chat1", "Chat13"),  
  input.data = data_all,  
  cmaes.results = model,  
  model.setup = model.setup  
)
```

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

Details

getParam
exported dataframe with parameters from fitted C-Q models

Value

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

`getResults`*Get Results*

Description

getResults of fitted C-Q models

Usage

```
getResults(cmaes.results = list(), model.setup = model.setup)
```

Arguments

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

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	<i>Get Stats</i>
----------	------------------

Description

getStats calculates the statistics for each fitted C-Q model

Usage

```
getStats(
  Chat.model.names = c("Chat1", "Chat13"),
  input.data = data_all,
  cmaes.results = models,
  model.setup = model.setup
)
```

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

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	<i>plot Results</i>
-------------	---------------------

Description

plotResults plots predictions from each fitted C-Q model

Usage

```
plotResults(
  Chat.model.names = c("Chat1", "Chat13"),
  input.data = data.frame(),
  model.setup = model.setup
)
```

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 AND predictions of concentration and baseflow
model.setup	lists of details about data, model, and site from setModels()

Details

plotResults
plots with parameters 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	<i>Run models</i>
-----------	-------------------

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	<i>Set models</i>
-----------	-------------------

Description

Set-up C-Q models for comparison

Usage

```
setModels(  
  Chat.model.names = c("Chat1", "Chat13"),  
  input.data = data_all,  
  Qthresh = 0,  
  Likelihood.name = "GaussLikelihood",  
  site.id = "",  
  site.name = ""  
)
```

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", "flow_mm_d")
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 ("GaussLikelihood", "GaussLikelihoodAR1", or "GaussLikelihoodAR3")
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

- * **getParam**
 - getParam, [3](#)
- * **getResults**
 - getResults, [3](#)
- * **getStats**
 - getStats, [4](#)
- * **package**
 - CQ2-package, [1](#)
- * **plotResults**
 - plotResults, [5](#)
- * **runModels**
 - runModels, [5](#)
- * **setModels**
 - setModels, [6](#)
- * **slow-quick**
 - CQ2-package, [1](#)
- * **water quality**
 - CQ2-package, [1](#)

CQ2 (CQ2-package), [1](#)
CQ2-package, [1](#)

getParam, [2](#), [3](#)
getResults, [2](#), [3](#)
getStats, [2](#), [4](#)

plotResults, [2](#), [5](#)

runModels, [2](#), [5](#)

setModels, [2](#), [6](#)