

Package: kwb.unicode (via r-universe)

August 30, 2024

Title Interface to Auto-Calibration Software UCODE

Version 0.1.0

Description Functions enabling to write UCODE input files and to run UCODE.

License MIT + file LICENSE

URL <https://github.com/KWB-R/kwb.unicode>

BugReports <https://github.com/KWB-R/kwb.unicode/issues>

Imports hydroGOF, kwb.wtaq, lattice

Remotes github::kwb-r/kwb.wtaq

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Repository <https://kwb-r.r-universe.dev>

RemoteUrl <https://github.com/KWB-R/kwb.unicode>

RemoteRef HEAD

RemoteSha daf2ec949c7d6c65c711481fd914cae91a86ba8

Contents

htmlForCalibrationHistory	2
htmlForCalibrationPage	3
htmlForCalibrationState	3
ucConf	4
ucConfGeneral	4
ucConfParameter	5
ucConf_Model_Command_Lines	6
ucConf_Model_Input_Files	7
ucConf_Model_Output_Files	7
ucConf_Observation_Data	7

ucConf_Parameter_Data	8
ucConf_Reg_GN_Controls	8
ucConf_UCODE_Control_Data	8
ucDefaultConfParameters	9
ucDefaultConfWeights	9
ucInpFileLines	10
ucPlotPng	10
ucPlotSummary	11
ucPrepCalib	11
ucReadSummary	13
ucRunParallel	14
ucSetPlaceholder	14
ucTable	15
ucTestBatchWtaq	16
ucWriteArgs	16
ucWriteBatchUcode	17
ucWriteBatchWtaq	17
ucWriteBatchWtaqParallel	18
ucWriteEvalGofR	18
ucWriteInstruction	19
ucWriteMainInput	19
ucWriteTemplate	20
uwGofEval	20
uwGofNames	21
uwGofTargetValue	21

Index	22
--------------	-----------

htmlForCalibrationHistory

History HTML for calibration progress visualisation

Description

History HTML for calibration progress visualisation

Usage

```
htmlForCalibrationHistory(n, current = "ucCurrent.html")
```

Arguments

n	number of plots
current	default: "ucCurrent.html"

htmlForCalibrationPage

Main HTML for calibration progress visualisation

Description

Main HTML for calibration progress visualisation

Usage

```
htmlForCalibrationPage(  
    htmlHist = "ucHistory.html",  
    htmlCurr = "ucCurrent.html"  
)
```

Arguments

htmlHist	default: "ucHistory.html"
htmlCurr	default: "ucCurrent.html"

htmlForCalibrationState

Content HTML for calibration progress visualisation

Description

Content HTML for calibration progress visualisation

Usage

```
htmlForCalibrationState(refresh = 1)
```

Arguments

refresh	refresh time in seconds
---------	-------------------------

ucConf	<i>Configure UCODE run</i>
--------	----------------------------

Description

Configure UCODE run

Usage

```
ucConf(
  general = ucConfGeneral(),
  parameters = ucDefaultConfParameters(),
  weights = ucDefaultConfWeights()
)
```

Arguments

general	configuration of control parameters as retrieved by ucConfGeneral .
parameters	data.frame with each row representing a parameter configuration as e.g. retrieved by ucConfParameter . For possible parameter names see ucSetPlaceholder
weights	Matrix of weights as e.g. retrieved by ucDefaultConfWeights . Matrix of weights is used when calculating overall fitness. Row names = names of fitness indicators out of the set of c("ME", "MAE", "MSE", "RMSE", "NRMSE", "PBIAS", "RSR", "rSD", "NSE", "mNSE", "rNSE", "d", "md", "rd", "cp", "r", "R2", "bR2", "KGE", "VE"). Column names = well names ("PW" = pumping well, observation wells named as in parameter setting). See example.

See Also

[ucPrepCalib](#)

ucConfGeneral	<i>General settings in UCODE configuration</i>
---------------	--

Description

General settings in UCODE configuration

Usage

```
ucConfGeneral(gof.digits = 5, maxIter = 100)
```

Arguments

gof.digits	Number of digits to be used in goodness of fit values. Default: 5
maxIter	Maximum number of parameter-estimation iterations allowed before stopping

See Also[ucConf](#)

ucConfParameter	<i>Create parameter for UCODE configuration</i>
-----------------	---

Description

Create parameter for UCODE configuration

Usage

```
ucConfParameter(
  paramname,
  startvalue = 1e+38,
  lowerValue = -1e+38,
  upperValue = 1e+38,
  lowerConstraint = 0,
  upperConstraint = 1,
  constrain = "yes",
  adjustable = "yes",
  maxChange = 2,
  perturbAmt = 0.5
)
```

Arguments

paramname	Parameter name (up to 12 characters; not case sensitive) - a character string that is used in a template file or in an equation of a derived parameter in the Derived_Parameters input block. Each parameter name needs to be unique and can not be the same as any parameter name defined in the Derived_Parameter input block. For possible parameter names see ucSetPlaceholder
startvalue	Starting parameter value. Default=A huge real number. The huge real number is obtained for the computer being used and commonly is about 1e38.
lowerValue	Smallest reasonable value for this parameter. Default= -(Huge real number). In absolute value, commonly about -1e38.
upperValue	Largest reasonable value for this parameter. Default= +(Huge real number). Commonly about +1e38.
lowerConstraint	Lower limit of considered parameter values. Default (HS): 0
upperConstraint	Upper limit of considered parameter values. Default (HS): 1
constrain	yes: constrain parameter values using LowerConstraint and UpperConstraint. Default (HS): yes.

adjustable	yes: change this value as needed depending on the purpose of the UCODE_2005 run defined in the UCODE_Control_Data input file. no: leave the value of this parameter unchanged. Default (HS): "yes".
maxChange	Maximum fractional parameter change allowed between parameter iterations. Default=2.0.
perturbAmt	Fractional amount of parameter value to perturb to calculate sensitivity. Commonly 0.01 to 0.10. Default (HS): 0.5. See discussion in Chapter 3.

Value

data.frame containing all argument values with column names equalling function argument names

See Also

[ucConf](#), [ucDefaultConfParameters](#)

Examples

```
# Use rbind to create a data.frame containing parameter information as
# required by ucConf:
cpara <- rbind(
  ucConfParameter(paramname = "HKR", startvalue = 2e-5,
    lowerConstraint = 1e-5, upperConstraint = 1e-1,
    maxChange = 1, perturbAmt = 0.5,
    constrain = "yes", adjustable = "yes"),
  ucConfParameter(paramname = "HKZ", startvalue = 2e-7,
    lowerConstraint = 1e-7, upperConstraint = 1e-3,
    maxChange = 1, perturbAmt = 0.5,
    constrain = "yes", adjustable = "yes"))

# set this parameter setting in default UCODE configuration
ucConf(parameters = cpara)
```

ucConf_Model_Command_Lines

ucConf Model Command Lines

Description

lines for input block *Model_Command_Lines*

Usage

```
ucConf_Model_Command_Lines(bfile)
```

Arguments

bfile name of batch file to be invoked by UCODE

 ucConf_Model_Input_Files

Configuration of Model Input Files

Description

Lines for input block *Model_Input_Files*

Usage

ucConf_Model_Input_Files()

ucConf_Model_Output_Files

Configuration of Model Output Files

Description

Lines for input block *Model_Output_Files*

Usage

ucConf_Model_Output_Files()

ucConf_Observation_Data

Configuration of Observation Data

Description

Lines for input block *Observation_Data*

Usage

ucConf_Observation_Data(weights)

Arguments

weights	Matix of weights used when calculating overall fitness. Row names = names of fitness indicators out of the set of c("ME", "MAE", "MSE", "RMSE", "NRMSE", "PBIAS", "RSR", "rSD", "NSE", "mNSE", "rNSE", "d", "md", "rd", "cp", "r", "R2", "bR2", "KGE", "VE"). Column names = well names ("PW" = pumping well, observation wells named as in parameter setting). For an example see ucConf
---------	---

ucConf_Parameter_Data *Configuration of Parameter Data*

Description

Lines for input block *Parameter_Data*

Usage

ucConf_Parameter_Data(confParameters)

Arguments

confParameters data frame representing table as required in *Parameter_Data* block of UCODE input file. , for example see [ucDefaultConfParameters](#)

ucConf_Reg_GN_Controls
ucConf Reg GN Controls

Description

lines for input block *UCODE_Control_Data*

Usage

ucConf_Reg_GN_Controls(maxIter)

Arguments

maxIter Maximum number of parameter-estimation iterations allowed before stopping

ucConf_UCODE_Control_Data
ucConf UCODE Control Data

Description

lines for input block *UCODE_Control_Data*

Usage

ucConf_UCODE_Control_Data()

 ucDefaultConfParameters

Default parameter definition for UCODE configuration

Description

Default parameter definition for UCODE configuration

Usage

```
ucDefaultConfParameters()
```

Value

data.frame with each row representing a model parameter

See Also

[ucConfParameter](#), [ucConf](#)

 ucDefaultConfWeights *Default weight matrix*

Description

Weights are given for each pair of well name (pumping well *PW* or observation wells) and name of goodness of fit (gof)-function.

Usage

```
ucDefaultConfWeights(
  wells = wtConfiguredWellnames(wtDefaultConfiguration()),
  gofs = c("RMSE", "NSE", "mNSE")
)
```

Arguments

wells	well names, default: wtConfiguredWellnames(wtDefaultConfiguration())
gofs	default: c("RMSE", "NSE", "mNSE")

Value

Matrix of weights used when calculating overall fitness. Row names = names of fitness indicators out of the set of c("ME", "MAE", "MSE", "RMSE", "NRMSE", "PBIAS", "RSR", "rSD", "NSE", "mNSE", "rNSE", "d", "md", "rd", "cp", "r", "R2", "bR2", "KGE", "VE"). Column names = well names ("PW" = pumping well, observation wells named as in parameter setting).

See Also[ucConf](#)

ucInpFileLines	<i>Input File Lines</i>
----------------	-------------------------

Description

Lines for UCODE main input file

Usage

```
ucInpFileLines(uconf, command)
```

Arguments

uconf	UCODE configuration as retrieved by ucConf
command	name of batch file that is invoked by UCODE with each iteration

ucPlotPng	<i>Plot WTAQ results to png file</i>
-----------	--------------------------------------

Description

Plot WTAQ results to png file

Usage

```
ucPlotPng(fname, wtaqResult, i)
```

Arguments

fname	fname
wtaqResult	wtaqResult
i	i

Value

???

ucPlotSummary	<i>Plot UCODE summary file</i>
---------------	--------------------------------

Description

Plot UCODE summary file

Usage

```
ucPlotSummary(sfile)
```

Arguments

sfile path to summary file

Value

???

ucPrepCalib	<i>Prepare calibration of WTAQ-model by creation of needed input and control files</i>
-------------	--

Description

All input files needed to run a UCODE calibration of a WTAQ model are created in the directory *tdir*. By default (*open.tdir = TRUE*) this directory is opened in the Windows explorer. You can use the created batch file `runParallelUCode.bat` to start the UCODE calibration.

Usage

```
ucPrepCalib(  
  wtaqConfiguration = wtDefaultConfiguration(),  
  uconf = ucConf(),  
  tdir = tempdir(),  
  open.tdir = TRUE,  
  wtaq = system.file("extdata", "wtaq.2.1.exe", package = "kwb.wtaq"),  
  ucode = system.file("extdata", "ucode_2005.exe", package = "kwb.wtaq")  
)
```

Arguments

wtaqConfiguration	WTAQ configuration as retrieved by wtConfigure
uconf	UCODE configuration as retrieved by ucConf
tdir	target directory in which all input and control files are to be created. Default: temporary directory of current R Session. Attention! This temporary directory will be emptied after the R session has finished!
open.tdir	if TRUE, the target directory will be opened in the Windows Explorer in order to allow inspecting the created files
wtaq	full path to WTAQ executable
ucode	full path to UCODE executable

See Also

[ucConf](#), [ucRunParallel](#)

Examples

```
# Define WTAQ configuration...
# Here: just load the configuration of WTAQ's Sample problem 2
wtaqConfiguration <- kwb.wtaq::wtConfigurationExample2()

# Define parameters
p <- rbind(
  ucConfParameter(
    "HKR",
    startvalue = 2e-5,
    lowerConstraint = 1e-5,
    upperConstraint = 1e-1,
    maxChange = 1,
    perturbAmt = 0.5
  ),
  ucConfParameter(
    "HKZ",
    startvalue = 2e-7,
    lowerConstraint = 1e-7,
    upperConstraint = 1e-3,
    maxChange = 1,
    perturbAmt = 0.5
  )
)

# Define weights
#
# The following weight matrix w gives a weight of 1 to the root mean square
# error (RMSE) and the Nash-Sutcliffe-Efficiency (NSE) between simulated
# and observed values at the pumping well ("PW") and to the NSE between
# simulated and observed values at the observation well "PS1".
#
# well -> PW PS1
```

```
w <- matrix(c( 1, 1, # RMSE <- GOF
              1, 1), # NSE
            nrow = 2, byrow = TRUE,
            dimnames = list(c("RMSE", "NSE"), c("PW", "PS1")))

# Create UCODE configuration
uconf <- ucConf(general = ucConfGeneral(maxIter = 100),
               parameters = p, weights = w)

# Create all input files needed to run the UCODE calibration of WTAQ
# sample problem 2
tdir <- ucPrepCalib(wtaqConfiguration = wtaqConfiguration, uconf = uconf)

# Run loop waiting for input files provided by UCODE (remove the comment
# character in the following line!)
#ucRunParallel(tdir, uconf = uconf)

# Now, run runParallelUCode.bat in the target directory...
```

ucReadSummary

Read UCODE summary file

Description

Read UCODE summary file

Usage

```
ucReadSummary(sfile)
```

Arguments

sfile path to summary file

Value

data frame, ordered by ITER

ucRunParallel *Run main calibration loop*

Description

Run main calibration loop

Usage

```
ucRunParallel(  
    uconf = ucConf(),  
    tdir = tempdir(),  
    wtaq = system.file("extdata", "wtaq.2.1.exe", package = "kwb.wtaq"),  
    FUN = NULL  
)
```

Arguments

uconf	UCode configuration as retrieved by ucConf
tdir	tdir
wtaq	wtaq
FUN	function that is invoked each time before WTAQ is run

Value

???

See Also

[ucPrepCalib](#)

ucSetPlaceholder *Set placeholder in WTAQ parameter setting*

Description

Set placeholder for parameter value in WTAQ parameter setting

Usage

```
ucSetPlaceholder(parname, wtaqConfiguration, pchar = "@")
```

Arguments

parname	Parameter name. Must be one of the aquifer parameters "bb", "hkr", "hkz", "ss", "sy" (for description see wtConfigureAquifer) or one of the drainage parameters "acc", "akk", "amm", "axmm" (for description see wtConfigureDrainage ; parameter "alpha" currently not supported) or one of the pumpwell parameters "qq", "rw", "rc", "zpd", "zpl", "sw" (for description see wtConfigurePumpwell) or one of the observation well parameters "r_i", "z1_i", "z2_i", "zp_i", "rp_i", "xll_i" (for description see wtConfigureObservationWell) where "i" is the number of the observation well according to the list index in the "obswell" section of the parameter setting.
wtaqConfiguration	WTAQ parameter setting as created by means of wtConfigure
pchar	character to be used as the very first and very last character of the placeholder. Default: "@"

Value

Parameter setting in which a parameter value is replaced by an appropriate placeholder

See Also

[ucConfParameter](#), [ucConf](#)

 ucTable

Compose Table Data

Description

Creates lines to be put in TABLE-data block of UCODE input file representing data in given data frame

Usage

```
ucTable(dat)
```

Arguments

dat	data frame
-----	------------

ucTestBatchWtaq	<i>Test run of batch file</i>
-----------------	-------------------------------

Description

Test run of batch file

Usage

```
ucTestBatchWtaq(bfile, wtaqConfiguration)
```

Arguments

bfile	path to batch file
wtaqConfiguration	WTAQ configuration

ucWriteArgs	<i>Write WTAQ arguments file</i>
-------------	----------------------------------

Description

Write WTAQ arguments file

Usage

```
ucWriteArgs(afile)
```

Arguments

afile	path to file containing arguments
-------	-----------------------------------

ucWriteBatchUcode	<i>Write batch file invoking UCODE</i>
-------------------	--

Description

Write batch file invoking UCODE

Usage

```
ucWriteBatchUcode(
  bfile,
  ucode = system.file("extdata", "ucode_2005.exe", package = "kwb.wtaq"),
  waitForInput = TRUE,
  ininfile = "ucode.in"
)
```

Arguments

bfile	full path to batch file
ucode	full path to UCODE executable file
waitForInput	if TRUE, "pause" is added to the batch file
ininfile	name of ucode input file

ucWriteBatchWtaq	<i>Write batch file invoked by UCODE, running first WTAQ and second the R script doing the goodness of fit evaluation</i>
------------------	---

Description

Write batch file invoked by UCODE, running first WTAQ and second the R script doing the goodness of fit evaluation

Usage

```
ucWriteBatchWtaq(bfile, wtaq, verbose, waitForInput = FALSE)
```

Arguments

bfile	path to batch file
wtaq	path to WTAQ executable
verbose	if TRUE, the output is not discarded
waitForInput	if TRUE, "pause" is added to the batch file

ucWriteBatchWtaqParallel

Write batch file invoked by UCODE, which is just waiting until WTAQ has finished and goodness of fit is available

Description

Write batch file invoked by UCODE, which is just waiting until WTAQ has finished and goodness of fit is available

Usage

```
ucWriteBatchWtaqParallel(bfile, recheckAfterMs = 10)
```

Arguments

bfile path to batchfile
recheckAfterMs time to wait for file, in milliseconds

ucWriteEvalGofR

Write R script evaluating goodness of fit from WTAQ output file

Description

Write R script evaluating goodness of fit from WTAQ output file

Usage

```
ucWriteEvalGofR(rfile, uconf)
```

Arguments

rfile path to R script file to be generated
uconf UCODE configuration

ucWriteInstruction	<i>Write UCODE instruction file, describing how to read goodness of fit values from wtaq.gof</i>
--------------------	--

Description

Write UCODE instruction file, describing how to read goodness of fit values from wtaq.gof

Usage

```
ucWriteInstruction(ifile, wtaqConfiguration, uconf)
```

Arguments

ifile	path to instruction file
wtaqConfiguration	WTAW configuration
uconf	UCODE configuration

ucWriteMainInput	<i>Write UCODE main input file</i>
------------------	------------------------------------

Description

Write UCODE main input file

Usage

```
ucWriteMainInput(ufile, uconf, command)
```

Arguments

ufile	path to UCODE input file
uconf	UCODE configuration
command	name of batch file that is invoked by UCODE with each iteration

ucWriteTemplate	<i>Write WTAQ input template file</i>
-----------------	---------------------------------------

Description

Write WTAQ input template file

Usage

```
ucWriteTemplate(tfile, wtaqConfiguration, uconf, pchar = "@")
```

Arguments

tfile	path to template file
wtaqConfiguration	WTAQ configuration
uconf	UCODE configuration
pchar	character indicating a placeholder

uwGofEval	<i>Evaluate goodness of fit for WTAQ result</i>
-----------	---

Description

Evaluate goodness of fit for WTAQ result

Usage

```
uwGofEval(res, logtimes = FALSE, gofs = names(uwGofTargetValue()), ...)
```

Arguments

res	either character string representing full path to WTAQ result plot file or data.frame containing the content of the WTAQ result plot file.
logtimes	if TRUE, time steps are supposed to be logarithmic, i.e. it is assumed that the result plot file is in “matrix form”.
gofs	vector of names of fitness functions, evaluated by <code>gof</code> . Default: <code>c("ME", "MAE", "MSE", "RMSE", "NRMSE", "PBIAS", "RSR", "rSD", "NSE", "mNSE", "rNSE", "d", "md", "rd", "cp", "r", "R2", "bR2", "KGE", "VE")</code>
...	arguments passed to <code>gof()</code> function, e.g. <code>digits</code> : decimal places used for rounding the goodness-of-fit indexes.

Value

Evaluation matrix

uwGofNames	<i>Names of available GOF functions</i>
------------	---

Description

Names of available *Goodness of fit (GOF)* functions

Usage

uwGofNames()

uwGofTargetValue	<i>Target values for different GOF functions</i>
------------------	--

Description

Target values for different GOF functions, i.e. values that GOF functions return if observed and calculated timeseries are identical

Usage

uwGofTargetValue(gofs = uwGofNames())

Arguments

gofs names of GOF functions of which target values are to be returned

Index

[gof](#), [20](#)

[htmlForCalibrationHistory](#), [2](#)
[htmlForCalibrationPage](#), [3](#)
[htmlForCalibrationState](#), [3](#)

[ucConf](#), [4](#), [5–7](#), [9](#), [10](#), [12](#), [14](#), [15](#)
[ucConf_Model_Command_Lines](#), [6](#)
[ucConf_Model_Input_Files](#), [7](#)
[ucConf_Model_Output_Files](#), [7](#)
[ucConf_Observation_Data](#), [7](#)
[ucConf_Parameter_Data](#), [8](#)
[ucConf_Reg_GN_Controls](#), [8](#)
[ucConf_UCODE_Control_Data](#), [8](#)
[ucConfGeneral](#), [4](#), [4](#)
[ucConfParameter](#), [4](#), [5](#), [9](#), [15](#)
[ucDefaultConfParameters](#), [6](#), [8](#), [9](#)
[ucDefaultConfWeights](#), [4](#), [9](#)
[ucInpFileLines](#), [10](#)
[ucPlotPng](#), [10](#)
[ucPlotSummary](#), [11](#)
[ucPrepCalib](#), [4](#), [11](#), [14](#)
[ucReadSummary](#), [13](#)
[ucRunParallel](#), [12](#), [14](#)
[ucSetPlaceholder](#), [4](#), [5](#), [14](#)
[ucTable](#), [15](#)
[ucTestBatchWtaq](#), [16](#)
[ucWriteArgs](#), [16](#)
[ucWriteBatchUcode](#), [17](#)
[ucWriteBatchWtaq](#), [17](#)
[ucWriteBatchWtaqParallel](#), [18](#)
[ucWriteEvalGofR](#), [18](#)
[ucWriteInstruction](#), [19](#)
[ucWriteMainInput](#), [19](#)
[ucWriteTemplate](#), [20](#)
[uwGofEval](#), [20](#)
[uwGofNames](#), [21](#)
[uwGofTargetValue](#), [21](#)

[wtConfigure](#), [12](#), [15](#)

[wtConfigureAquifer](#), [15](#)
[wtConfigureDrainage](#), [15](#)
[wtConfigureObservationWell](#), [15](#)
[wtConfigurePumpwell](#), [15](#)