

Package: DM (via r-universe)

January 27, 2025

Type Package

Title Dynamic Model

Version 1.7.5

Date 2017-03-27

Depends R (>= 2.15.0), R2jags, R2WinBUGS, coda, KernSmooth, GenSA,
stringr, tools, knitr, shiny, openxlsx, VRAP (>= 1.3.6),
Gmedian, rmarkdown

SystemRequirements WinBUGS 1.4, jags (>= 3.3.0)

Author Eli Holmes, Martin Liermann, Howard Coleman, NOAA, Seattle, USA

Maintainer Elizabeth Holmes - NOAA Federal <eli.holmes@noaa.gov>

Description This package uses comma-delimited data exported from the
A&P Excel files to estimate the SR parameters for a
user-specified SR function.

License GPL-2

LazyData yes

BuildVignettes yes

NeedsCompilation no

RoxygenNote 6.0.1

VignetteBuilder knitr

Config/pak/sysreqs make jags libicu-dev zlib1g-dev

Repository <https://nmfs-opensci.r-universe.dev>

RemoteUrl <https://github.com/nwfsc-math-bio/DM>

RemoteRef HEAD

RemoteSha b8feb2e17aeb74abe83c83edadd50aa282ab1e66

Contents

calcHarvest	2
calcInitVals	3

calculateHarvested	3
calculatePEandSmsy	4
centerProdCap	4
conditionalcenterProdCap	5
createBUGSdata	6
createPriors	6
createRAVfile	7
DM	9
findOptimum	9
getPostDraws	10
info	10
plotAgeComp	11
plotCapacity	11
plotCoefs	12
plotPriors	12
plotProductivity	13
plotResults	13
readData	14
runJAGS	15
runModel	16
selectSR	17
simplePlot	17
srfuctions	18
SSTfuncDM	19
summarizeRuns	19
uncenterProdCap	20
writeBUGScode	21
writeReport	21
writeResultsToFile	22
Index	23

calcHarvest	<i>Calculate harvest rate</i>
-------------	-------------------------------

Description

Calculate the harvest rate by year and age of return

Usage

```
calcHarvest(bdat)
```

Arguments

bdat	data for the bayesian specification of a DM run
------	---

Value

Harvest rate (scalar).

calcInitVals	<i>Calculate initial values for MCMC.</i>
--------------	---

Description

calcInitVals computes initial values for MCMC algorithm used to compute the posterior distribution for the model parameters.

Usage

```
calcInitVals(mlEst, dat, input, bdat, sims = NULL)
```

Arguments

mlEst	The maximum likelihood estimate for the four SR parameters: log(prod), log(capacity), log(msCoef), log(flowCoef). Output from findOptimum().
dat	data from the A & P file
input	a list with the other values needed for a DM run. The following are examples naturalMort, analysisType = "DM", SRfunction, includeMarineSurvival, includeFlow
bdat	The in a format appropriate for JAGS implementation of the model.
sims	MCMC simulations from a previous fit. If sims=NULL all initial values are based on estimates derived from the data.

Value

A list with initial values for prod (production), logCap (log capacity), msCoef (the coefficient for marine survival), flowCoef (the coefficient for stream survival), rMu and maturationRate_logit.

calculateHarvested	<i>Calculate harvested recruits and escapement</i>
--------------------	--

Description

Calculate harvested recruits and escapement

Usage

```
calculateHarvested(r1)
```

Arguments

r1 a single run list output by the runModel function

Value

A list with harvested recruits and escapement numbers.

calculatePEandSmsy *Calculate process error and SMSY.*

Description

This function takes input and dat produced from the A & P files along with x (the posteriors) from getPosteriors(). It computes the process error estimates (MSE and autoCorr or mean and sterr) along with the SMSY for each posterior draw. It then assembles all these into a dataframe tDat and returns that.

Usage

```
calculatePEandSmsy(input, dat, postdraws)
```

Arguments

input a list with the other values needed for a DM run. The following are examples naturalMort, analysisType = "DM", SRfunction, includeMarineSurvival, includeFlow

dat data from the A & P file

postdraws posterior list from getPostDraws() call

Value

A dataframe tDat with a column for posterior draws from each parameter and the smsy for each draw.

centerProdCap *Center productivity and log capacity parameters*

Description

Center capacity and productivity so that parameters can be compared to estimates from a Bayesian analysis where the covariates were centered.

Usage

```
centerProdCap(input, dat, params)
```

Arguments

dat	data from the A & P file
params	a vector of parameters with names flowCoef, msCoef, prod, cap. Assumed to be uncentered, e.g. the ML estimates from the SR model with covariates.

Value

a vector of centered parameters with names flowCoef, msCoef, prod, cap.

conditionalcenterProdCap

Center productivity and log capacity parameters conditioned on values in input

Description

Center capacity and productivity so that parameters can be compared to estimates from an analysis where the covariates were (possibly) centered. VRAP uses the uncentered parameters. The input\$centerFlow and input\$centerMS variables specify whether and which covariates were centered for the Bayesian analysis.

Usage

```
conditionalcenterProdCap(input, dat, params)
```

Arguments

dat	data from the A & P file
params	a vector of parameters with names flowCoef, msCoef, prod, cap. Assumed to be uncentered, e.g. the ML estimates from the SR model with covariates.

Value

a vector of centered parameters with names flowCoef, msCoef, prod, cap.

createBUGSdata *Create bugs data for model with or without age data.*

Description

This function creates bugs data (bdat).

Usage

```
createBUGSdata(dat, input, priors = NULL, allData = FALSE)
```

Arguments

dat	data from the A & P file
input	a list with the other values needed for a DM run. The following are examples naturalMort, analysisType = "DM" or "SS", SRfunction, includeMarineSurvival, includeFlow
priors	the parameters for the prior distributions used in the JAGS models.
allData	if allData=TRUE the data for both model types is returned. If allData=FALSE, just the data necessary for the specified model is returned.

Value

dataframe returns a list with the data and priors formatted as input for the JAGS model.

createPriors *Create a priors list.*

Usage

```
createPriors(pMode = 20, pSig = 7, pMin = 0, pMax = 40, cMu = 9,
  cSig = 50, cMin = 3, cMax = 12, msMu = 1e-04, msSig = 10,
  msMin = 1e-04, msMax = 1000, flowMu = -1e-04, flowSig = 10,
  flowMin = -1000, flowMax = -1e-04)
```

Arguments

pMode	Mode of the production prior (dlnorm)
pSig	Sigma of the production prior
pMin	Lowerbound of the production prior
pMax	Upperbound of the production prior
cMu	Mean of the log capacity prior (dnorm)
cSig	Sigma of the log capacity prior

cMin	Lowerbound of the log capacity prior
cMax	Upperbound of the log capacity prior
msMu	Mean of the marine survival coef prior (dnorm)
msSig	Sigma of the marine survival coef prior
msMin	Lowerbound of the marine survival coefficient prior
msMax	Upperbound of the marine survival coefficient prior
flowMu	Mean of the marine survival coef prior (dnorm)
flowSig	Sigma of the marine survival coef prior
flowMin	Lowerbound of the flow coefficient prior; the coefficient is always negative
flowMax	Upperbound of the flow coefficient prior; the coefficient is always negative

Value

a list with parameters for each prior.

createRAVfile	<i>Create rav file.</i>
---------------	-------------------------

Description

This function takes the output from the DM model and creates a rav file that can be used for running the VRAP simulations. The parameter simInd indicates which MCMC sim to use. This takes the information from the input file in the DM spreadsheet (read in by readDMData) and the results of the MCMC simulations. Current version has no uncertainty in management error and base exploitation rate must be included as a parameter. This could be calculated with the SR params, maturation rates, natMort, and harvest.

Usage

```
createRAVfile(bdat, input, tDat, dat, filename = "temp_rav.rav",
  estType = "median", sim = 1, rav.options = list())
```

Arguments

bdat	data for the bayesian specification of a DM run
input	a list with the other values needed for a DM run. The following are examples naturalMort, analysisType = "DM", SRfunction, includeMarineSurvival, includeFlow
tDat	a data frame of the transformed posteriors and calculated process error and Smsy for each draw. From calculatePEandSmsy(). This is specifically for VRAP input. The productivity (a) and capacity (b) posteriors are not adjusted (centered) if the covariates do not have 0 mean. VRAP needs the SR parameters in this raw form using non-centered (de-meaned) covariates.
dat	data from the A & P file

filename	name to give the rav file
estType	("median")
sim	indicates which MCMC sim from the posteriors to use if estType is not median
rav.options	list of the rav file options to use that do not come from the posteriors. randomSeed (0) numRuns (1000) The number of simulations to run for computing probabilities in VRAP. numYears (25) The number of years to project forward in the simulations. minAge (2) maxAge (5) convergeCrit (0.001) centerCovFlag ("NO") "NO"/"YES" marineSurvType ("Autoc") "Autoc"/"Cycle"/"Trend" flowType ("Autoc") "Autoc"/"Cycle"/"Trend" modelDepensation ("NO") "NO"/"YES" depensation (300) QETcritical (63) dependsPar3 (1) recruitsFromAdultSpawners ("YES") "NO"/"YES" SRvariation ("YES") "NO"/"YES" smoltToAdultVar ("NO") "NO"/"YES" baseExploitationRate (0.67) includeManagementError ("YES") "NO"/"YES" manageErrorA (65.3946) derived from estimates from Puget Sound for all except STL and WRS manageErrorB (0.0158) derived from estimates from Puget Sound for all except STL and WRS lowerEscThreshold (200) numYearsToAvg (5) runType ("ER") "ER"/"Pop" bufferMin (0)

Value

nothing is returned but the rav file is written.

DM	<i>Shiny app to run DM and VRAP</i>
----	-------------------------------------

Description

This function takes the input from an A & P files, runs DM to estimate SR parameters and saves a .rav file and .RData file. The .rav file can be downloaded or directly used for running the VRAP simulations by clicking the VRAP tab.

Usage

```
DM()
```

findOptimum	<i>Optimization functions.</i>
-------------	--------------------------------

Description

Because the likelihood surfaces for these problems often has a severe banana shape with a poorly defined maximum finding an optimum is often non-trivial. If an optimum is given in the input file I use that as an initial value for nlm otherwise, I found that nlm often gets stuck in a local min. So here I have iterated between a genetic optimization algorithm and nlm. This works more often but still at times misses the optimum. This is adhoc.

The covariates are log-transformed and centered if `input$centerMS=TRUE` or `input$centerFlow=TRUE`.

Usage

```
findOptimum(dat, input, silent = FALSE)
```

Arguments

dat	data from the A & P file
input	a list with the other values needed for a DM run.
silent	(TRUE/FALSE)

Details

In `SRFunctions()`, `bev-holt` is defined as $S/(S*\exp(-p[2])+\exp(-p[1]))*\exp(p[3]*\log MS)*\exp(p[4]*\log Flow)$
 In `DM (writeBUGSmodel.R)`, $R = [S/(S/\exp(\log Cap) + (1/prod))] \exp(marineInd*\log MS) \exp(flowCoef*\log Flow)$
 so `p[1] = log(prod)`, constrained to be positive `p[2] = log(cap)`, constrained to be positive `p[3] = msCoef`, `p[4] = flowCoef`

Value

A list. `$estimate` parameters at the minimum sum of squared residuals. The parameters are `prod`, `cap`, `msCoef`, `flowCoef`. `$value` is the sum of squared residuals at the minimum.

getPostDraws	<i>Get the simulations from a jags output list.</i>
--------------	---

Usage

```
getPostDraws(dmObj)
```

Arguments

dmObj	a saved DM object (list) from runModel() with the objects "input", "dat", "result", "bdat" and "tDat"
-------	---

Value

a list of vectors of the posterior draws for each parameter.

info	<i>Helper file for creating text for the reports.</i>
------	---

Description

This uses data in the RData file from a saved DM run to create text describing the model fit.

Usage

```
info(RData.file)
```

Arguments

RData.file	a saved DM run as a RData file with the objects "input", "dat", "result", and "tDat"
------------	--

Value

a list with text strings.

plotAgeComp	<i>Create plots of age composition</i>
-------------	--

Description

This function is called by plotResults(). Age composition data versus calendar year. The years with age composition data are represented with a stacked barplot. With age 2 on the bottom (black) and age 5 on the top (white). The numbers above each bar represent the age composition sample size.

Usage

```
plotAgeComp(dmObj, proportions = TRUE)
```

Arguments

dmObj	a saved DM object (list) from runModel() with the objects "input", "dat", "result", "bdat" and "tDat"
proportions	If proportions=TRUE, the age composition proportions are plotted. If proportions=FALSE, the number of samples in each age are plotted.

Value

The plot is made.

plotCapacity	<i>Create plots of capacity posteriors</i>
--------------	--

Description

Capacity posterior distribution. This is the posterior and prior distributions for the spawner-recruit capacity parameter.

Usage

```
plotCapacity(dmObj, xLab = "Capacity parameter", plotType = "histogram",
  binSize = NULL, xLims = NULL)
```

Arguments

dmObj	a saved DM object (list) from runModel() with the objects "input", "dat", "result", "bdat" and "tDat"
xLab	x-axis label
plotType	("histogram")
binSize	The width of the bin if the plot type is histogram.
xLims	x-axis limits

Value

The plot is made.

plotCoefs	<i>Create plots of covariate coefficients</i>
-----------	---

Usage

```
plotCoefs(dmObj, xLim = c(-3, 3), xLabs = c("Marine Survival coefficient",
      "Flow coefficient"))
```

Arguments

dmObj	a saved DM object (list) from runModel() with the objects "input", "dat", "result", "bdat" and "tDat"
xLim	x-axis limits
xLabs	x-axis labels

Value

The plot is made.

plotPriors	<i>Create plots of priors for productivity and log capacity.</i>
------------	--

Description

This function is used by plotResults()

Usage

```
plotPriors(priors)
```

Arguments

priors	List of priors parameters. Created with createPriors().
--------	---

Value

The plot is made.

plotProductivity *Create plots of productivity posteriors*

Usage

```
plotProductivity(dmObj, xLab = "Productivity parameter",
  plotType = "histogram", binSize = NULL, xLims = NULL)
```

Arguments

dmObj	a saved DM object (list) from runModel() with the objects "input","dat","result", "bdat" and "tDat"
xLab	x-axis label
plotType	("histogram")
binSize	The width of the bin if the plot type is histogram.
xLims	x-axis limits

Value

The plot is made.

plotResults *Create plots from DM runs*

Description

Create several plots based on the DM model runs. This calls the functions for various plots: plot-Priors, plotPosteriors, plotAgeComp, plotCoefs

Usage

```
plotResults(dmObj, plotType = c("a.and.p.data", "priors", "SR", "recruits",
  "covariates", "harvest", "EscData", "EscAndAge", "escapement", "barPlot",
  "ageComp"), plotDest = "default", plotName = "tmp")
```

Arguments

dmObj	a saved DM object (list) from runModel() with the objects "input","dat","result", and "tDat"
plotType	The plots and order in which to make them covariates harvest EscData

	EscAndAge
	escapement
	SR
	recruits recruit residuals, recruits/spawners #and effective harvest on recruits
	barPlot a bar plot showing spawners and recruits broken down by pHOS and harvested
	ageComp
	priors Priors used in the DM model
	a.and.p.data Raw data from the A and P input file.
plotDest	Plot destination. Can be "screen", "png", "default", "none". If "screen", separate graphics windows are opened. If "none", nothing is plotted but the figure captions are returned.
plotName	prefix to apply to filename for generated plots if plotDest="png"

Value

The plot is saved to a file or plotted to screen or default device depending on plotDest. A vector of plot captions is returned invisibly.

readData	<i>This reads in the csv file saved from an A & P file and creates dat and input</i>
----------	--

Description

The DM tab of the A & P excel file has all the necessary inputs to run DM. This reads that .csv file and makes a list needed for the DM functions. creates input which has the other user specified values for the rav file.

Usage

```
readData(a.and.p.file, input, folder = "./", silent = FALSE)
```

Arguments

a.and.p.file	a csv file saved from the DM tab of an A & P excel file
input	a list with the other values needed for a DM run. The following are the defaults used, but can be passed in to specify something different. population name of the population. Defaults to using value in A and P file. naturalMort vector of natural mortality for age 1:6. Defaults to c(0.5,0.4,0.3,0.2,0.1,0) firstYear First year of spawner data to use. Defaults to using value in cell C4 on DynamicsInput tab in A and P file. lastYear Last year of spawner data to use. Defaults to using value in cell C5 on DynamicsInput tab in A and P file.

MSYfirstYear Defaults to using firstYear.

MSYlastYear Defaults to using lastYear.

analysisType "DM" or "SS". Defaults to DM.

SRfunction ("ricker")/"bevertonHolt"/"hockeyStick": spawner-recruit function

includeMarineSurvival "yes"/("no"): include marine survival covariate

includeFlow "yes"/("no"): include flow covariate

initialPopSize Initial population size for age 2 to 5. Defaults to using values in cells N6:Q6 on DynamicsInput tab in A and P file.

prod Used for initial conditions of optimizers and MCMC algorithm. Default is NA which means the ML estimates are used as the initial conditions.

cap Used for initial conditions of optimizers and MCMC algorithm. Default is NA which means the ML estimates are used as the initial conditions.

msCoef Used for initial conditions of optimizers and MCMC algorithm. Default is NA which means the ML estimates are used as the initial conditions.

flowCoef Used for initial conditions of optimizers and MCMC algorithm. Default is NA which means the ML estimates are used as the initial conditions.

centerMS Set mean of the MS covariate to zero in the SR function. Default is TRUE.

centerFlow Set mean of the flow covariate to zero in the SR function. Default is TRUE.

escapementObsSD NULL

age2correction Correction for seeing fewer age 2 fish when sampling for age composition. The value to pass in is an estimate of detection probability for age 2 fish / detection probability for ages 3-5 fish. Default is 1.0: no correction.

silent Whether to print progress messages.

Value

The result is a list with dat and input for the DM functions.

runJAGS

Call JAGS to get the posterior draws.

Description

This function calls jags to get the posterior draws.

Usage

```
runJAGS(bdat, calcInits, sims = 10000, numChains = 3, details = FALSE)
```

Arguments

bdat	a list of containing the data necessary for the JAGS model
calcInits	a list or function that generates a list of initial values for the JAGS model
sims	the number of MCMC simulations used for each chain after a burnin of 10 percent of sims
numChains	number of MCMC chains to use
details	if details=TRUE, more detailed results are returned.

Value

If details=FALSE, a list of MCMC chains for the monitored model parameters is returned. If details=TRUE, a model object generated by the jags function is returned (this contains the simulations along with additional information)

runModel	<i>Run the DM model</i>
----------	-------------------------

Description

This function takes the list created by readDMData() and runs the Bayesian model to estimate the SR parameters. The output is written to a .csv file and an .Rdata file. It writes a file, mod1.txt to the user's working directory

Usage

```
runModel(filename, input = NULL, priors = NULL, run = TRUE,
         sims = 10000, numChains = 3, oldSims = NULL, silent = FALSE)
```

Arguments

filename	csv file saved from the DM tab of A & P file. Must have age data.
input	A list that includes parameters for running the model (see readData for details) of particular interest are the parameters SRfunction, analysisType, and escapementObsSD. SRfunction sets the spawner-recruit function to be used. Current options are "ricker", "bevertonHolt" or "hockeyStick". analysisType can be either "DM" or "SS". DM corresponds to the traditional Dynamic Model which uses the maturation rates from the DynamicsInput tab in the A&P table. SS corresponds to the State Space model which estimates the maturation rates and includes observation error in the spawner counts. If escapementObsSD is a positive real number, observation error is fixed at that value. If it is NULL, observation error is a free parameter that is estimated.
priors	parameters for prior distributions in the JAGS model. See createPriors .
run	if run=TRUE, the model is run and the results returned, otherwise all of the information necessary to run the model is returned without running the model

sims	length of each MCMC chain; default is 10000
numChains	number of MCMC chains. default is 3
oldSims	?
silent	Whether to print progress messages.

Value

a dmObj: a list of output for a single model fit.

selectSR	<i>Helper function that takes the name of SR function and returns the recruits</i>
----------	--

Usage

```
selectSR(x)
```

Arguments

x name of SR function as a text string: "ricker", "bevertonHolt", "hockeyStick"

Value

a function

simplePlot	<i>Create posterior credible interval plots from multiple DM runs</i>
------------	---

Usage

```
simplePlot(modellist, modNames = names(modellist)[modOrder],
  paramList = c("obsSD", "procSD", "prod", "logCap"), funcList = NULL,
  cols = 4, leftMargin = 12, modOrder = 1:length(modellist))
```

Arguments

modellist	a list of multiple run objects (dmObj) based on different models. Each run object in the list is from one call of runModel
modNames	more detailed names of models. If the runs are named in the list, it uses those by default.
paramList	The parameters you want to compare from the model output.
funcList	functions that take the dmObj as their only parameter return a posterior distribution you want to compare across models. the user can make their own functions or utilize one of the available functions (procVarAutoCorr, and SmsyFunc)
cols	The number of columns of figures in the plot.
leftMargin	the size of the left figure margin (passed to oma in par).
modOrder	the order of the models. defaults to the the order in the list.

Value

The plot is made.

srfuctions

SR fucitons

Description

Takes the S, LOG flow, LOG marine survival covariates and p parameters and returns the recruits based on the specified SR function: ricker(), bevertonHolt(), hockeyStick(). Note that it assumes that the log of the covariates are input. All covariates appear in the form $\exp(p \cdot \text{cov})$. $p[1]=\log a$; $p[2]=\log b$. The other p are the covariate coefficients.

Usage

```
ricker(S, covariates, p)
```

```
bevertonHolt(S, covariates, p)
```

```
hockeyStick(S, covariates, p)
```

Arguments

S	spawner count
covariates	A list with the covariates. The ms covariate is normally the first and the flow covariate is the second. Though order does not matter as the p's are matched based on order the covariate appears in the list.
p	the SR function a, b, and the coefficients for each covariate in the covariate list. All covariates appear in the form $\exp(\text{exp}(p) \cdot \text{cov})$. $p[1]=a$; $p[2]=b$. The other p match the covariates.

Value

recruits (scalar)

recruits from ricker: $S \cdot \exp(p[1]) \cdot \exp(-S/\exp(p[2])) \cdot \exp(p[4] \cdot \text{flow}) \cdot \exp(p[3] \cdot \text{marineInd})$

recruits from beverton Holt: $S / (S \cdot \exp(-p[2]) + \exp(-p[1])) \cdot \exp(p[4] \cdot \text{flow}) \cdot \exp(p[3] \cdot \text{marineInd})$

recruits from hockey stick: $\text{ifelse}(\exp(p[1]) \cdot S < \exp(p[2]), \exp(p[1]) \cdot S, \exp(p[2])) \cdot \exp(p[4] \cdot \text{flow}) \cdot \exp(p[3] \cdot \text{marineInd})$

SSTfuncDM	<i>Original dynamic model</i>
-----------	-------------------------------

Description

Implements the DM model from the original RER process. Estimates escapement by year using escapement data from previous years, maturation rates, harvest rates, and rates of natural mortality. The estimated escapement is log transformed and subtracted from the log transformed values of observed escapement. These differences are then squared and summed to get the sum of squares residual which is returned by the function. This function can be used to estimate the spawner-recruit parameters through the use of an optimization function.

Usage

```
SSTfuncDM(p, SRfunc, dat, input, returnAll = FALSE)
```

Arguments

p	The values of the SR function parameters. p[1] and p[2] are logged to keep them positive. p[1]=log(prod), p[2]=log(cap). The other p depend on what covariates are in the model. These are unlogged as they could be positive or negative depending on the covariate.. The length of p is 2 + number of covariates. If there are 2 covariates, then p[3]= msCoef and p[4]=flowCoef.
SRfunc	a SR function: ricker, bevertonHolt, hockeyStick, as returned by selectSR()
dat	data from the A & P file
input	a list with the other values needed for a DM run. The following are examples naturalMort, analysisType = "DM", SRfunction, covariates, includeMarineSurvival, includeFlow
returnAll	(TRUE/FALSE)

Value

Returns the sum of squares residuals between the observed and estimated escapement. If returnAll is true, the estimated and observed escapement are also returned.

summarizeRuns	<i>Summarize runs</i>
---------------	-----------------------

Description

Summarize DM runs

Usage

```
summarizeRuns(runList)
```

Arguments

runList a list of multiple run objects (dmObj). Each run object in the list is from one call of runModel

Value

a data frame with summary statistics for each run

uncenterProdCap	<i>Uncenter productivity and log capacity parameters</i>
-----------------	--

Description

Transform capacity and productivity estimates to what they would be if the covariates were centered during estimation (Bayesian or ML). VRAP uses the uncentered parameters. Note, if the coefficients were centered during estimation, input\$centerMS=TRUE and/or input\$centerFlow=TRUE. These values are used to determine whether to apply the correction. Thus input is assumed to match the parameter values in params.

Usage

```
uncenterProdCap(input, dat, params)
```

Arguments

dat data from the A & P file

params a vector of parameters with names flowCoef, msCoef, prod, cap. Assumed to be from the runModel() Bayesian analysis or findOptimum() call, so centered (or not) based on input\$centerFlow and input\$centerMS

Value

a vector of uncentered parameters with names flowCoef, msCoef, prod, cap.

See Also

centerProdCap

writeBUGScore	<i>write the model in BUGS language</i>
---------------	---

Description

create a mod1.txt which contains the model in the BUGS language

Usage

```
writeBUGScore(input = NULL, outputText = FALSE)
```

Arguments

input	a list with the other values needed for a DM run. The following are examples naturalMort, analysisType = "DM", SRfunction, covariates, includeMarineSurvival, includeFlow
outputText	(TRUE/FALSE)

Details

Note that the flow coefficient is always negative and prior set as a normal with upper (negative bound)

Value

nothing but writes the file mod1.txt which is read in to run the BUGS model

writeReport	<i>Creates a report with plots.</i>
-------------	-------------------------------------

Usage

```
writeReport(dmObj = NULL, dmObj.RData.file = NULL,
  output.file = "report1", rav.options = list(),
  output.format = c("pdf_document"), output.dir = getwd(),
  input.type = "xRnw")
```

Arguments

dmObj	a saved DM object (list) from runModel() with the objects "input", "dat", "result", and "tDat"
dmObj.RData.file	a saved DM object as a RData file with the objects "input", "dat", "result", and "tDat"
output.file	Filename to give saved report.

<code>rav.options</code>	list of the rav file options to use that do not come from the posteriors.
<code>output.format</code>	list of output formats. If <code>input.type=="xRmd"</code> , any format that <code>rmarkdown::render</code> allows is fine. If <code>input.type=="xRnw"</code> , only "pdf_document" is allowed.
<code>output.dir</code>	Directory where to save the output files. Defaults to working directory.
<code>input.type</code>	xRmd for rmarkdown. xRnw for knitr.

Value

Nothing. The report is written to a PDF and tex file.

`writeResultsToFile` *Save DM output.*

Description

Write DM output (posteriors) to csv file and RData file.

Usage

```
writeResultsToFile(input, dat, tDat, mlEst, bdat, plist, filename)
```

Arguments

<code>input</code>	a list with the other values needed for a DM run. The following are the defaults used, but can be passed in to specify something different: <code>naturalMort = c(0.5,0.4,0.3,0.2,0.1,0)</code> , <code>analysisType = "DM"</code> , <code>SRfunction = "ricker"</code> , <code>covariates = "no"</code> , <code>includeMarineSurvival = "no"</code> , <code>includeFlow = "no"</code>
<code>dat</code>	data from the A & P file
<code>tDat</code>	a data frame of the posteriors and calculated process error and Smsy for each draw. From <code>calculatePEandSmsy()</code> .
<code>mlEst</code>	The maximum likelihood estimate for the four SR parameters
<code>bdat</code>	data for the bayesian specification of a DM run
<code>filename</code>	name to give the outputfiles file
<code>population</code>	Name of the population.

Value

nothing is returned but the csv and RData files are written.

Index

bevertonHolt (srfunctions), 18

calcHarvest, 2

calcInitVals, 3

calculateHarvested, 3

calculatePEandSmsy, 4

centerProdCap, 4

conditionalcenterProdCap, 5

createBUGSdata, 6

createPriors, 6, 16

createRAVfile, 7

DM, 9

DMVRAP (DM), 9

findOptimum, 9

getPostDraws, 10

hockeyStick (srfunctions), 18

info, 10

plotAgeComp, 11

plotCapacity, 11

plotCoefs, 12

plotPriors, 12

plotProductivity, 13

plotResults, 13

readData, 14

ricker (srfunctions), 18

runJAGS, 15

runModel, 16

selectSR, 17

simplePlot, 17

srfunctions, 18

SSTfuncDM, 19

summarizeRuns, 19

uncenterProdCap, 20

writeBUGScode, 21

writeReport, 21

writeResultsToFile, 22