Package 'MSEtool'

January 10, 2020

Type Package

Title Management Strategy Evaluation Toolkit

Version 1.4.3

Date 2020-01-09

Maintainer Tom Carruthers <t.carruthers@oceans.ubc.ca>

Description Simulation tools for management strategy evaluation are provided for the 'DLMtool' operating model to inform data-rich fisheries.
'MSEtool' provides complementary assessment models of varying complexity with standardized reporting, diagnostic tools for evaluating
assessment models within closed-loop simulation, and helper functions for building more complex operating models and management procedures.

License GPL-3

Depends R (>= 3.3.0), DLMtool (>= 5.3.1)

Imports MASS, TMB, coda, corpcor, dplyr, gplots, grDevices, graphics,
methods, mvtnorm, reshape2, snowfall, stats, utils, abind,
rmarkdown

LinkingTo TMB, RcppEigen

LazyData yes

LazyLoad yes

RoxygenNote 7.0.2

Suggests knitr, testthat, r4ss, shiny

VignetteBuilder knitr

URL http://www.datalimitedtoolkit.org

BugReports https://github.com/tcarruth/MSEtool/issues

NeedsCompilation yes

Author Quang Huynh [aut],
Tom Carruthers [aut, cre],
Adrian Hordyk [aut],
Chris Grandin [ctb] (iSCAM functions)

Repository CRAN

Date/Publication 2020-01-10 13:20:02 UTC

R topics documented:

MSEtool-package *Management Strategy Evaluation*

Description

Simulation tools for management strategy evaluation are provided for the DLMtool operating model to inform data-rich fisheries. MSEtool provides complementary assessment models of varying complexity with standardized reporting, diagnostic tools for evaluating assessment models within closed-loop simulation, and helper functions for building more complex operating models and management procedures.

How to use MSEtool

The main features of MSEtool are the assessment models and the ability to make management procedures by combining assessment models with harvest control rules. Such MPs can be used and tested in management strategy evaluation with DLMtool operating models. An overview of these features is available in the MSEtool vignette.

The following assessment models are available:

- Surplus production (SP, SP_SS, SP_Fox, and `spict`)
- Delay difference (DD, cDD, DD_SS, and cDD_SS)
- Statistical catch-at-age (SCA, SCA2, and SCA_Pope)
- Virtual population analysis (VPA)

MSEtool also contains multiMSE, a platform for multi-stock and multi-fleet operating models based on components from DLMtool. An overview of multiMSE is available in the multiMSE vignette.

All MSEtool vignettes can also be viewed by typing `browseVignettes("MSEtool")` into the R console or through the MSEtool webpage on CRAN.

Additional links

See the DLMtool User Guide for a detailed description of how to use the DLMtool package.

See the Data-Limited Toolkit Website for more information on DLMtool, including an interactive demo of the main features of the toolkit, information on case studies where the toolkit has been applied, and more about the history and development of the DLMtool.

Author(s)

Quang Huynh <q.huynh@oceans.ubc.ca>

Tom Carruthers <t.carruthers@oceans.ubc.ca>

Adrian Hordyk <a.hordyk@oceans.ubc.ca>

References

Carruthers, T.R., Punt, A.E., Walters, C.J., MacCall, A., McAllister, M.K., Dick, E.J., Cope, J. 2014. Evaluating methods for setting catch limits in data-limited fisheries. Fisheries Research. 153: 48-68.

Carruthers, T.R., Kell, L.T., Butterworth, D.S., Maunder, M.N., Geromont, H.F., Walters, C., McAllister, M.K., Hillary, R., Levontin, P., Kitakado, T., Davies, C.R. Performance review of simple management procedures. ICES Journal of Marine Science. 73: 464-482.

`Albacore_TwoFleet` *A two-fleet Albacore operating model*

Description

A generic operating model for an albacore stock with two fishing fleets. The first fleet has dome-shaped selectivity (similar to a baitboat fleet) while the second fleet exhibits logistic selectivity (such as a longline fleet). With the `CatchFrac` slot, we generate a 30 ratio between the baitboat-longline fleets in the most recent historical year.

Usage

```
Albacore_TwoFleet
```

Format

An object of class MOM.

Examples

```
## Plot historical effort and selectivity between the 2 fleets
plot(Albacore_TwoFleet)

## Generate data (e.g., catch, length comps) from the fleets
Hist <- multiMSE(Albacore_TwoFleet, Hist = TRUE)
DataList <- Hist$Data

# Fleet one
summary(DataList[[1]][[1]])

# Fleet two
summary(DataList[[1]][[2]])
```

Assessment-class *Class*-Assessment

Description

An S4 class that contains assessment output. Created from a function of class Assess.

Slots

Model Name of the assessment model.

Name Name of Data object.

conv Logical. Whether the assessment model converged (defined by whether TMB returned a positive-definite covariance matrix for the model).

UMSY Estimate of exploitation at maximum sustainable yield.

FMSY Estimate of instantaneous fishing mortality rate at maximum sustainable yield.

MSY Estimate of maximum sustainable yield.

BMSY Biomass at maximum sustainable yield.

SSBMSY Spawning stock biomass at maximum sustainable yield.

VBMSY Vulnerable biomass at maximum sustainable yield.

B0 Biomass at unfished equilibrium.

R0 Recruitment at unfished equilibrium.

N0 Abundance at unfished equilibrium.

SSB0 Spawning stock biomass at unfished equilibrium.

VB0 Vulnerable biomass at unfished equilibrium.

h Steepness.

U Time series of exploitation.

U_UMSY Time series of relative exploitation.

FMort Time series of instantaneous fishing mortality.

F_FMSY Time series of fishing mortality relative to MSY.

B Time series of biomass.

B_BMSY Time series of biomass relative to MSY.

B_B0 Time series of depletion.

SSB Time series of spawning stock biomass.

SSB_SSBMSY Time series of spawning stock biomass relative to MSY.

SSB_SSB0 Time series of spawning stock depletion.

VB Time series of vulnerable biomass.

VB_VBMSY Time series of vulnerable biomass relative to MSY.

VB_VB0 Time series of vulnerable biomass depletion.

`R` Time series of recruitment.

`N` Time series of population abundance.

`N_at_age` Time series of numbers-at-age matrix.

`Selectivity` Selectivity-at-age matrix.

`Obs_Catch` Observed catch.

`Obs_Index` Observed index.

`Obs_C_at_age` Observed catch-at-age matrix.

`Catch` Predicted catch.

`Index` Predicted index.

`C_at_age` Predicted catch-at-age matrix.

`Dev` A vector of estimated deviation parameters.

`Dev_type` A description of the deviation parameters, e.g. "log recruitment deviations".

`NLL` Negative log-likelihood. A vector for the total likelihood, integrated across random effects if applicable, components, and penalty term (applied when `U > 0.975` in any year).

`SE_UMSY` Standard error of UMSY estimate.

`SE_FMSY` Standard error of FMSY estimate.

`SE_MSY` Standard error of MSY estimate.

`SE_U_UMSY_final` Standard error of U/UMSY in the terminal year.

`SE_F_FMSY_final` Standard error of F/FMSY in the terminal year.

`SE_B_BMSY_final` Standard error of B/BMSY in the terminal year.

`SE_B_B0_final` Standard error of B/B0 in the terminal year.

`SE_SSB_SSBMSY_final` Standard error of SSB/SSBMSY in the terminal year.

`SE_SSB_SSB0_final` Standard error of SSB/SSB0 in the terminal year.

`SE_VB_VBMSY_final` Standard error of VB/VBMSY in the terminal year.

`SE_VB_VB0_final` Standard error of VB/VB0 in the terminal year.

`SE_Dev` A vector of standard errors of the deviation parameters.

`info` A list containing the data and starting values of estimated parameters for the assessment.

`obj` A list with components returned from `MakeADFun`.

`opt` A list with components from calling `nlminb` to `obj`.

`SD` A list (class sdreport) with parameter estimates and their standard errors, obtained from `sdreport`.

`TMB_report` A list of model output reported from the TMB executable, i.e. `obj$report()`, and derived quantities (e.g. MSY).

`dependencies` A character string of data types required for the assessment.

Author(s)

Q. Huynh

See Also

plot.Assessment summary.Assessment retrospective profile_likelihood make_MP

Examples

```
output <- DD_TMB(Data = DLMtool::Red_snapper)
class(output)
```

`avail` *What objects of this class are available*

Description

Generic class finder

Usage

```
avail(classy, all_avail = TRUE)
```

Arguments

`classy`	A class of object (character string, e.g. 'Fleet')
`all_avail`	Logical. If TRUE, function will return all objects of class `classy` available to user. If FALSE, returns only those objects included in MSEtool.

Details

Finds objects of the specified class in the global environment or in the MSEtool and DLMtool packages. This function is an addendum to the `avail` function in DLMtool.

Author(s)

Q. Huynh

Examples

```
avail("Assess")
avail("HCR")
avail("Stock")
avail("MP")
avail("MP", all_avail = FALSE)
```

Awatea2OM — *Reads MCMC estimates from Awatea (Paul Starr) processed r file structure into an operating model*

Description

A function that uses the file location of a fitted Awatea model post-processed into a set of rmd files

Usage

```
Awatea2OM(
  AwateaDir,
  nsim = 48,
  proyears = 50,
  Name = NULL,
  Source = "No source provided",
  Author = "No author provided",
  verbose = T
)
```

Arguments

AwateaDir	A folder with Awatea files
nsim	The number of simulations to take for parameters with uncertainty (for OM@cpars custom parameters)
proyears	The number of projection years for MSE
Name	The name of the operating model
Source	Reference to assessment documentation e.g. a url
Author	Who did the assessment
verbose	Should the r4ss function SS_ouput return detailed messages?

Author(s)

T. Carruthers

CASAL2OM — *Reads MLE estimates from CASAL file structure into an operating model*

Description

A (prototype) function that uses the file location of a fitted CASAL assessment model including input files to population the various slots of an operating model with MLE parameter estimates. The function mainly populates the Stock and Fleet portions of the operating model; the user still needs to parameterize most of the observation and implementation portions of the operating model.

Usage

```
CASAL2OM(
  CASALdir,
  Obs = DLMtool::Precise_Unbiased,
  Imp = DLMtool::Perfect_Imp,
  Name = NA,
  Agency = NA,
  Region = NA,
  Sponsor = NA,
  Latitude = NA,
  Longitude = NA,
  nsim = 48,
  proyears = 50,
  interval = 4,
  pstar = 0.5,
  maxF = 2,
  reps = 1,
  seed = 1,
  Common_Name = NA,
  Species = NA,
  Source = NA,
  Author = NA
)
```

Arguments

CASALdir	A folder with CASAL input and output files in it
Obs	The observation model (class Obs).
Imp	The implementation model (class Imp).
Name	The common name of the operating model
Agency	The fishery management agency
Region	The geographical location
Sponsor	Who funded the work
Latitude	In degrees north
Longitude	In degrees west
nsim	The number of simulations to take for parameters with uncertainty (for OM@cpars custom parameters).
proyears	The number of projection years for MSE
interval	The number of years between management updates
pstar	The quantile for TAC management given stochasticity
maxF	The maximum allowable F in the operating model.
reps	The number of stochastic replicates within each simulation in the operating model.
seed	The random seed for the operating model.

Common_Name	The name of the species
Species	The species latin name
Source	Reference to assessment documentation e.g. a url
Author	Who did the assessment

Value

An object of class OM.

Author(s)

T. Carruthers

See Also

SS2OM

CASALpars *Rips MLE estimates from CASAL file structure*

Description

A function that uses the file location of a fitted CASAL assessment model including input files to extract data required to populate an OMx class operating model.

Usage

```
CASALpars(CASALdir)
```

Arguments

CASALdir	A folder with Stock Synthesis input and output files in it

Value

A list.

Author(s)

T. Carruthers

See Also

CASAL2OM

cDD *Continuous Delay-differential assessment model*

Description

A catch and index-based assessment model. Compared to the discrete delay-difference (annual time-step in production and fishing), the delay-differential model (cDD) is based on continuous recruitment and fishing mortality within a time-step. The continuous model works much better for populations with high turnover (e.g. high F or M, continuous reproduction). This model is conditioned on catch and fits to the observed index. In the state-space version (cDD_SS), recruitment deviations from the stock-recruit relationship are estimated.

Usage

```
cDD(
  x = 1,
  Data,
  SR = c("BH", "Ricker"),
  rescale = "mean1",
  start = NULL,
  fix_h = TRUE,
  fix_F_equilibrium = TRUE,
  n_itF = 5L,
  silent = TRUE,
  opt_hess = FALSE,
  n_restart = ifelse(opt_hess, 0, 1),
  control = list(iter.max = 5000, eval.max = 10000),
  ...
)

cDD_SS(
  x = 1,
  Data,
  SR = c("BH", "Ricker"),
  rescale = "mean1",
  start = NULL,
  fix_h = TRUE,
  fix_F_equilibrium = TRUE,
  fix_sigma = FALSE,
  fix_tau = TRUE,
  n_itF = 5L,
  integrate = FALSE,
  silent = TRUE,
  opt_hess = FALSE,
  n_restart = ifelse(opt_hess, 0, 1),
  control = list(iter.max = 5000, eval.max = 10000),
  inner.control = list(),
```

```
  ...
)
```

Arguments

x	An index for the objects in Data when running in closed loop simulation. Otherwise, equals to 1 when running an assessment.
Data	An object of class Data.
SR	Stock-recruit function (either "BH" for Beverton-Holt or "Ricker").
rescale	A multiplicative factor that rescales the catch in the assessment model, which can improve convergence. By default, "mean1" scales the catch so that time series mean is 1, otherwise a numeric. Output is re-converted back to original units.
start	Optional list of starting values. Entries can be expressions that are evaluated in the function. See details.
fix_h	Logical, whether to fix steepness to value in Data@steep in the assessment model.
fix_F_equilibrium	Logical, whether the equilibrium F prior to the first year of the model is estimated. If TRUE, F_equilibruim is fixed to value provided in start (if provided), otherwise, equal to zero (assumes unfished conditions).
n_itF	Integer, the number of iterations to solve F conditional on the observed catch.
silent	Logical, passed to MakeADFun, whether TMB will print trace information during optimization. Used for dignostics for model convergence.
opt_hess	Logical, whether the hessian function will be passed to nlminb during optimization (this generally reduces the number of iterations to convergence, but is memory and time intensive and does not guarantee an increase in convergence rate). Ignored if integrate = TRUE.
n_restart	The number of restarts (calls to nlminb) in the optimization procedure, so long as the model hasn't converged. The optimization continues from the parameters from the previous (re)start.
control	A named list of parameters regarding optimization to be passed to nlminb.
...	Additional arguments (not currently used).
fix_sigma	Logical, whether the standard deviation of the index is fixed. If TRUE, sigma is fixed to value provided in start (if provided), otherwise, value based on Data@CV_Ind.
fix_tau	Logical, the standard deviation of the recruitment deviations is fixed. If TRUE, tau is fixed to value provided in start (if provided), otherwise, equal to 1.
integrate	Logical, whether the likelihood of the model integrates over the likelihood of the recruitment deviations (thus, treating it as a state-space variable). Otherwise, recruitment deviations are penalized parameters.
inner.control	A named list of arguments for optimization of the random effects, which is passed on to newton via MakeADFun.

Details

To provide starting values for cDD, a named list can be provided for R0 (unfished recruitment) and and h (steepness) via the start argument (see example).

For cDD_SS, additional start values can be provided for and sigma and tau, the standard deviation of the index and recruitment variability, respectively.

Value

An object of Assessment containing objects and output from TMB.

Required Data

- cDD: Cat, Ind, Mort, L50, vbK, vbLinf, vbt0, wla, wlb, MaxAge
- cDD_SS: Cat, Ind, Mort, L50, vbK, vbLinf, vbt0, wla, wlb, MaxAge

Optional Data

- cDD: steep
- cDD_SS: steep, CV_Ind, sigmaR

Author(s)

Q. Huynh

References

Hilborn, R., and Walters, C., 1992. Quantitative Fisheries Stock Assessment: Choice, Dynamics and Uncertainty. Chapman and Hall, New York.

See Also

DD_TMB plot.Assessment summary.Assessment retrospective profile make_MP

Examples

```
#### Observation-error delay difference model
res <- cDD(Data = DLMtool::Red_snapper)

# Provide starting values
start <- list(R0 = 1, h = 0.95)
res <- cDD(Data = DLMtool::Red_snapper, start = start)

summary(res@SD) # Parameter estimates

### State-space version
### Set recruitment variability SD = 0.6 (since fix_tau = TRUE)
res <- cDD_SS(Data = Red_snapper, start = list(tau = 0.6))
```

compare_models *Compare output from several assessment models*

Description

Plot biomass, recruitment, and fishing mortality time series from several . This function can be used to compare outputs among different assessment models from the same Data object.

Usage

```
compare_models(..., label = NULL, color = NULL)
```

Arguments

...	Objects of class Assessment.
label	A character vector of the models for the legend.
color	A vector of colors for each assessment model.

Value

A set of figures of biomass, recruitment, and fishing mortality estimates among the models.

Author(s)

Q. Huynh

Examples

```
res <- cDD_SS(Data = DLMtool::SimulatedData)
res2 <- SCA(Data = DLMtool::SimulatedData)
res3 <- SCA2(Data = DLMtool::SimulatedData)
res4 <- VPA(Data = DLMtool::SimulatedData)

compare_models(res, res2, res3)
```

Data-rich-MP *Data-rich management procedures*

Description

A suite of data-rich management procedures (MPs) included in the package. Additional MPs, with specific model configurations (e.g., stock-recruit function or fixing certain parameters) or alternative ramped harvest control rules can be created with make_MP and the available Assess and HCR objects.

Usage

```
SCA_MSY(x, Data, reps = 1)

SCA_75MSY(x, Data, reps = 1)

SCA_4010(x, Data, reps = 1)

DDSS_MSY(x, Data, reps = 1)

DDSS_75MSY(x, Data, reps = 1)

DDSS_4010(x, Data, reps = 1)

SP_MSY(x, Data, reps = 1)

SP_75MSY(x, Data, reps = 1)

SP_4010(x, Data, reps = 1)
```

Arguments

x	A position in the Data object.
Data	An object of class Data
reps	Numeric, the number of stochastic replicates for the management advice.

Value

An object of class Rec which contains the management recommendation.

Functions

- SCA_MSY: A statistical catch-at-age model with a TAC recommendation based on fishing at UMSY, and default arguments for configuring SCA.
- SCA_75MSY: An SCA with a TAC recommendation based on fishing at 75% of UMSY.
- SCA_4010: An SCA with a 40-10 control rule.
- DDSS_MSY: A state-space delay difference model with a TAC recommendation based on fishing at UMSY, and default arguments for configuring DD_SS.
- DDSS_75MSY: A state-space delay difference model with a TAC recommendation based on fishing at 75% of UMSY.
- DDSS_4010: A state-space delay difference model with a 40-10 control rule.
- SP_MSY: A surplus production model with a TAC recommendation based on fishing at UMSY, and default arguments for configuring SP.
- SP_75MSY: A surplus production model with a TAC recommendation based on fishing at 75% of UMSY.
- SP_4010: A surplus production model with a 40-10 control rule.

Examples

```
avail("MP", all_avail = FALSE)

## Not run:
myMSE <- DLMtool::runMSE(DLMtool::testOM, MPs = c("FMSYref", "SCA_MSY", "SCA_4010"))

## End(Not run)
```

DD_TMB *Delay - Difference Stock Assessment in TMB*

Description

A simple delay-difference assessment model using a time-series of catches and a relative abundance index and coded in TMB. The model is conditioned on effort and estimates predicted catch. In the state-space version, recruitment deviations from the stock-recruit relationship are estimated.

Usage

```
DD_TMB(
  x = 1,
  Data,
  SR = c("BH", "Ricker"),
  rescale = "mean1",
  start = NULL,
  fix_h = TRUE,
  fix_U_equilibrium = TRUE,
  silent = TRUE,
  opt_hess = FALSE,
  n_restart = ifelse(opt_hess, 0, 1),
  control = list(iter.max = 5000, eval.max = 10000),
  ...
)

DD_SS(
  x = 1,
  Data,
  SR = c("BH", "Ricker"),
  rescale = "mean1",
  start = NULL,
  fix_h = TRUE,
  fix_U_equilibrium = TRUE,
  fix_omega = FALSE,
  fix_tau = TRUE,
  integrate = FALSE,
  silent = TRUE,
  opt_hess = FALSE,
```

```
  n_restart = ifelse(opt_hess, 0, 1),
  control = list(iter.max = 5000, eval.max = 10000),
  inner.control = list(),
  ...
)
```

Arguments

x	An index for the objects in Data when running in closed loop simulation. Otherwise, equals to 1 when running an assessment.
Data	An object of class Data.
SR	Stock-recruit function (either "BH" for Beverton-Holt or "Ricker").
rescale	A multiplicative factor that rescales the catch in the assessment model, which can improve convergence. By default, "mean1" scales the catch so that time series mean is 1, otherwise a numeric. Output is re-converted back to original units.
start	Optional list of starting values. Entries can be expressions that are evaluated in the function. See details.
fix_h	Logical, whether to fix steepness to value in Data@steep in the assessment model.
fix_U_equilibrium	Logical, whether the equilibrium harvest rate prior to the first year of the model is estimated. If TRUE, U_equilibruim is fixed to value provided in start (if provided), otherwise, equal to zero (assumes virgin conditions).
silent	Logical, passed to MakeADFun, whether TMB will print trace information during optimization. Used for dignostics for model convergence.
opt_hess	Logical, whether the hessian function will be passed to nlminb during optimization (this generally reduces the number of iterations to convergence, but is memory and time intensive and does not guarantee an increase in convergence rate). Ignored if integrate = TRUE.
n_restart	The number of restarts (calls to nlminb) in the optimization procedure, so long as the model hasn't converged. The optimization continues from the parameters from the previous (re)start.
control	A named list of parameters regarding optimization to be passed to nlminb.
...	Additional arguments (not currently used).
fix_omega	Logical, whether the standard deviation of the catch is fixed. If TRUE, omega is fixed to value provided in start (if provided), otherwise, value based on Data@CV_Cat.
fix_tau	Logical, the standard deviation of the recruitment deviations is fixed. If TRUE, tau is fixed to value provided in start (if provided), otherwise, equal to 1.
integrate	Logical, whether the likelihood of the model integrates over the likelihood of the recruitment deviations (thus, treating it as a random effects/state-space variable). Otherwise, recruitment deviations are penalized parameters.
inner.control	A named list of arguments for optimization of the random effects, which is passed on to newton via MakeADFun.

Details

To provide starting values for DD_TMB, a named list can be provided for R0 (virgin recruitment), h (steepness), and q (catchability coefficient) via the start argument (see example).

For DD_SS, additional start values can be provided for and omega and tau, the standard deviation of the catch and recruitment variability, respectively.

Value

An object of Assessment containing objects and output from TMB.

Functions

- DD_TMB: Observation-error only model
- DD_SS: State-Space version of Delay-Difference model

Required Data

- DD_TMB: Cat, Ind, Mort, L50, vbK, vbLinf, vbt0, wla, wlb, MaxAge
- DD_SS: Cat, Ind, Mort, L50, vbK, vbLinf, vbt0, wla, wlb, MaxAge

Optional Data

- DD_TMB: steep
- DD_SS: steep, CV_Cat

Note

Similar to many other assessment models, the model depends on assumptions such as stationary productivity and proportionality between the abundance index and real abundance. Unsurprisingly the extent to which these assumptions are violated tends to be the biggest driver of performance for this method.

Author(s)

T. Carruthers & Z. Siders. Zach Siders coded the TMB function.

References

Carruthers, T, Walters, C.J,, and McAllister, M.K. 2012. Evaluating methods that classify fisheries stock status using only fisheries catch data. Fisheries Research 119-120:66-79.

Hilborn, R., and Walters, C., 1992. Quantitative Fisheries Stock Assessment: Choice, Dynamics and Uncertainty. Chapman and Hall, New York.

See Also

plot.Assessment summary.Assessment retrospective profile make_MP

Examples

```
#### Observation-error delay difference model
res <- DD_TMB(Data = DLMtool::Red_snapper)

# Provide starting values
start <- list(R0 = 1, h = 0.95)
res <- DD_TMB(Data = DLMtool::Red_snapper, start = start)

summary(res@SD) # Parameter estimates

### State-space version
### Set recruitment variability SD = 0.3 (since fix_tau = TRUE)
res <- DD_SS(Data = Red_snapper, start = list(tau = 0.3))
```

`diagnostic_AM` *diagnostic_AM (diagnostic of Assessments in MSE): Did Assess models converge during MSE?*

Description

Diagnostic check for convergence of Assess models during MSE. Assess models write output to the DLMenv environment if the MP was created with make_MP with argument `diagnostic = TRUE`. This function summarizes and plots the diagnostic information.

Usage

```
diagnostic_AM(MSE, MP = NULL, gradient_threshold = 0.1, figure = TRUE)
```

Arguments

`MSE`	An object of class MSE created by `runMSE`. If no MSE object is available, use argument `MP` instead.
`MP`	A character vector of MPs with assessment models.
`gradient_threshold`	The maximum magnitude (absolute value) desired for the gradient of the likelihood.
`figure`	Logical, whether a figure will be drawn.

Value

A matrix with diagnostic performance of assessment models in the MSE. If `figure = TRUE`, a set of figures: traffic light (red/green) plots indicating whether the model converged (defined if a positive-definite Hessian matrix was obtained), the optimizer reached pre-specified iteration limits (as passed to `nlminb`), and the maximum gradient of the likelihood in each assessment run. Also includes the number of optimization iterations function evaluations reported by `nlminb` for each application of the assessment model.

Author(s)

Q. Huynh

See Also

retrospective_AM

Examples

```
## Not run:
DD_MSY <- make_MP(DD_TMB, HCR_MSY, diagnostic = "min")
show(DD_MSY)

##### Ensure that PPD = TRUE in runMSE function
myMSE <- runMSE(DLMtool::testOM, MPs = "DD_MSY", PPD = TRUE)
diagnostic_AM(myMSE)

## End(Not run)
```

`expandHerm` *Expand the Herm list in SexPars to a matrix of fractions at age*

Description

Expand the Herm list in SexPars to a matrix of fractions at age

Usage

```
expandHerm(Herm, maxage, np, nsim)
```

Arguments

`Herm`	A list of Hermaphroditic fractions at age (starting age class 1)
`maxage`	The maximum age of stocks being simulated
`np`	The total number of stocks being simulated
`nsim`	The number of simulations

Author(s)

T. Carruthers

fetch.file.names *Reads iSCAM Data, Control and Projection files*

Description

A function for returning the three types of iSCAM input and output files

Usage

```
fetch.file.names(path, filename)
```

Arguments

path	File path
filename	The filename

Author(s)

Chris Grandin (DFO PBS)

getinds *Characterize posterior predictive data*

Description

Characterize posterior predictive data

Usage

```
getinds(
  PPD,
  styr,
  res = 6,
  tsd = c("Cat", "Cat", "Cat", "Ind", "ML"),
  stat = c("slp", "AAV", "mu", "slp", "slp")
)
```

Arguments

PPD	An object of class Data stored in the Misc slot of an MSE object following a call of runMSE(PPD = TRUE).
styr	Positive integer, the starting year for calculation of quantities
res	Positive integer, the temporal resolution (chunks - normally years) over which to calculate quantities

tsd	Character vector of names of types of data: Cat = catch, Ind = relative abundance index, ML = mean length in catches
stat	Character vector of types of quantity to be calculated: slp = slope(log(x)), AAV = average annual variability, mu = mean(log(x))

Value

A 3D array of results (type of data/stat (e.g. mean catches),time period (chunk), simulation)

Author(s)

T. Carruthers

References

Carruthers and Hordyk 2018

---

getnIVs *Count independent variables for a MICE relationship at position x in a Rel list*

Description

Count independent variables for a MICE relationship at position x in a Rel list

Usage

```
getnIVs(x, Rel)
```

Arguments

x	Position of a MICE relationship in the list Rel (MOM@Rel)
Rel	The list of MICE relationships (MOM@Rel)

Author(s)

T.Carruthers

HCRlin *Generic linear harvest control rule based on biomass*

Description

A general function used by HCR_ramp that adjusts the TAC by a linear ramp based on estimated biomass.

Usage

```
HCRlin(Brel, LRP, TRP, rel_min = 0, rel_max = 1)
```

Arguments

Brel	Improper fraction: An estimate of biomass (either absolute or relative, e.g. B/BMSY or B/B0).
LRP	Improper fraction: the Limit Reference Point, the biomass below which the adjustment is at its minimum, e.g. zero, no fishing. Same units as Brel.
TRP	Improper fraction: the Target Reference Point, the biomass above which the adjustment is at its maximum. Same units as Brel.
rel_min	The relative maximum value (e.g. a multiple of FMSY) if Brel < LRP.
rel_max	The relative maximum value (e.g. a multiple of FMSY) if Brel > TRP.

Value

a TAC or TAE adjustment factor.

Author(s)

T. Carruthers

Examples

```
#40-10 linear ramp
Brel <- seq(0, 1, length.out = 200)
plot(Brel, HCRlin(Brel, 0.1, 0.4), xlab = "Estimated B/B0", ylab = "Relative change in F",
main = "A 40-10 harvest control rule", type = 'l', col = 'blue')
abline(v = c(0.1,0.4), col = 'red', lty = 2)
```

HCR_FB *A Harvest Control Rule using B/BMSY and F/FMSY to adjust TAC or TAE.*

Description

A Harvest Control Rule using B/BMSY and F/FMSY to adjust TAC or TAE.

Usage

```
HCR_FB(Brel, Frel, Bpow = 2, Bgrad = 1, Fpow = 1, Fgrad = 1)
```

Arguments

Brel	improper fraction: an estimate of Biomass relative to BMSY
Frel	improper fraction: an estimate of Fishing mortality rate relative to FMSY
Bpow	non-negative real number: controls the shape of the biomass adjustment, when zero there is no adjustment
Bgrad	non-negative real number: controls the gradient of the biomass adjustment
Fpow	non-negative real number: controls the adjustment speed relative to F/FMSY. When set to 1, next recommendation is FMSY. When less than 1 next recommendation is between current F and FMSY.
Fgrad	improper fraction: target Fishing rate relative to FMSY

Value

a TAC or TAE adjustment factor.

Author(s)

T. Carruthers

References

Made up for this package

Examples

```
res <- 100
Frel <- seq(1/2, 2, length.out = res)
Brel <- seq(0.05, 2, length.out=res)
adj <- array(HCR_FB(Brel[rep(1:res, res)], Frel[rep(1:res, each = res)],
                    Bpow = 2, Bgrad = 1, Fpow = 1, Fgrad = 0.75), c(res, res))
contour(Brel, Frel, adj, nlevels = 20, xlab = "B/BMSY", ylab = "F/FMSY",
        main = "FBsurface TAC adjustment factor")
abline(h = 1, col = 'red', lty = 2)
abline(v = 1, col = 'red', lty = 2)
legend('topright', c("Bpow = 2", "Bgrad = 1", "Fpow = 1", "Fgrad = 0.75"), text.col = 'blue')
```

HCR_MSY *Harvest control rule to fish at some fraction of maximum sustainable yield*

Description

A simple control rule that specifies the total allowable catch (TAC) to be the product of current vulnerable biomass and UMSY.

Usage

```
HCR_MSY(Assessment, reps = 1, MSY_frac = 1, ...)
```

Arguments

Assessment	An object of class Assessment with estimates of FMSY or UMSY and vulnerable biomass in terminal year.
reps	The number of stochastic samples of the TAC recommendation.
MSY_frac	The fraction of FMSY or UMSY for calculating the TAC (e.g. MSY_frac = 0.75 fishes at 75% of FMSY).
...	Miscellaneous arguments.

Value

An object of class Rec with the TAC recommendation.

Author(s)

Q. Huynh

References

Punt, A. E, Dorn, M. W., and Haltuch, M. A. 2008. Evaluation of threshold management strategies for groundfish off the U.S. West Coast. Fisheries Research 94:251-266.

See Also

make_MP HCR_ramp

Examples

```
# create an MP to run in closed-loop MSE (fishes at UMSY)
DD_MSY <- make_MP(DD_TMB, HCR_MSY)
class(DD_MSY)

# The same MP which fishes at 75% of UMSY
DD_75MSY <- make_MP(DD_TMB, HCR_MSY, MSY_frac = 0.75)
class(DD_MSY)
```

```
## Not run:
myOM <- DLMtool::runMSE(DLMtool::testOM, MPs = c("FMSYref", "DD_MSY"))

## End(Not run)
```

HCR_ramp *Linearly ramped harvest control rules*

Description

An output control rule with a ramp that reduces the TAC recommendation linearly with respect to fishing mortality (F) or harvest rate (U) when the relative biomass (i.e., spawning depletion or spawning biomass relative to that at MSY) is less than the target reference point (TRP). The TAC reduction is linearly reduced with respect to F to a minimum value when the relative biomass is less than the limit reference point (LRP). For example, the TRP and LRP for spawning depletion is 0.4 and 0.1, respectively, in the 40-10 control rule. Class HCR objects are typically used with function make_MP.

Usage

```
HCR_ramp(
  Assessment,
  reps = 1,
  LRP,
  TRP,
  rel_min = 0,
  rel_max = 1,
  RP_type = c("SSB_SSB0", "SSB_SSBMSY"),
  ...
)

HCR40_10(Assessment, reps = 1, ...)

HCR60_20(Assessment, reps = 1, ...)
```

Arguments

Assessment	An object of class Assessment with estimates of FMSY or UMSY, vulnerable biomass, and spawning biomass depletion in terminal year.
reps	The number of stochastic samples of the TAC recommendation.
LRP	Numeric, the limit reference point.
TRP	Numeric, the target reference point.
rel_min	The relative maximum value (e.g. a multiple of FMSY) if Brel < LRP.
rel_max	The relative maximum value (e.g. a multiple of FMSY) if Brel > TRP.
RP_type	The reference point metric for TRP and LRP ("SSB_SSB0" for spawning depletion by default, or "SSB_SSBMSY" for spawning biomass relative to MSY).
...	Miscellaneous arguments.

Details

HCR_ramp is the generic ramped-HCR function where user specifies LRP, TRP, and relative biomass metric, as well as minimum and maximum values for adjusting the fishing mortality.

HCR40_10 is a common U.S. west coast control rule (LRP and TRP of 0.1 and 0.4 spawning depletion, respectively), while HCR60_20 is more conservative than 40-10, with LRP and TRP of 0.2 and 0.6 spawning depletion, respectively).

Value

An object of class Rec with the TAC recommendation.

Author(s)

Q. Huynh & T. Carruthers

References

Deroba, J.J. and Bence, J.R. 2008. A review of harvest policies: Understanding relative performance of control rules. Fisheries Research 94:210-223.

Edwards, C.T.T. and Dankel, D.J. (eds.). 2016. Management Science in Fisheries: an introduction to simulation methods. Routledge, New York, NY. 460 pp.

Punt, A. E, Dorn, M. W., and Haltuch, M. A. 2008. Evaluation of threshold management strategies for groundfish off the U.S. West Coast. Fisheries Research 94:251-266.

Restrepo, V.R. and Power, J.E. 1999. Precautionary control rules in US fisheries management: specification and performance. ICES Journal of Marine Science 56:846-852.

See Also

HCR_MSY HCRlin make_MP

Examples

```
# 40-10 linear ramp
Brel <- seq(0, 1, length.out = 200)
plot(Brel, HCRlin(Brel, 0.1, 0.4), xlab = "Estimated SSB/SSB0",
ylab = "Prescribed F relative to FMSY", main = "40-10 harvest control rule",
type = "l", col = "blue")
abline(v = c(0.1, 0.4), col = "red", lty = 2)

# create a 40-10 MP to run in closed-loop MSE
DD_40_10 <- make_MP(DD_TMB, HCR40_10)

# Alternatively,
DD_40_10 <- make_MP(DD_TMB, HCR_ramp, LRP = 0.1, TRP = 0.4)

# An SCA with LRP and TRP at 0.4 and 0.8, respectively, of SSB/SSBMSY
SCA_80_40 <- make_MP(SCA, HCR_ramp, LRP = 0.4, TRP = 0.8, RP_type = "SSB_SSBMSY")

# A conservative HCR that fishes at 75% of FMSY at B > 80% BMSY but only reduces F
```

```
# to 10% of FMSY if B < 40% BMSY.
SCA_conservative <- make_MP(SCA, HCR_ramp, LRP = 0.4, TRP = 0.8, rel_max = 0.75,
rel_min = 0.1, RP_type = "SSB_SSBMSY")

# Figure of this conservative HCR
Brel <- seq(0, 1, length.out = 200)
Frel <- HCRlin(Brel, 0.4, 0.8, rel_max = 0.75, rel_min = 0.1)
plot(Brel, Frel, xlab = "Estimated SSB/SSB_MSY", ylab = "Prescribed F relative to FMSY",
type = "l", col = "blue")
abline(v = c(0.4, 0.8), col = "red", lty = 2)

## Not run:
myOM <- DLMtool::runMSE(DLMtool::testOM, MPs = c("FMSYref", "DD_40_10"))

## End(Not run)
```

`iSCAM2Data` *Reads data from iSCAM file structure into a DLMtool Data object*

Description

A function that uses the file location of a fitted iSCAM model including input files to population the various slots of an data object. iSCAM2OM relies on several functions written by Chris Grandin (DFO PBS).

Usage

```
iSCAM2Data(
  iSCAMdir,
  Name = NULL,
  Source = "No source provided",
  length_timestep = 1,
  Author = "No author provided"
)
```

Arguments

`iSCAMdir`	A folder with iSCAM input and output files in it
`Name`	The name of the operating model
`Source`	Reference to assessment documentation e.g. a url
`length_timestep`	How long is a model time step in years (e.g. a quarterly model is 0.25, a monthly model 1/12)
`Author`	Who did the assessment

Author(s)

T. Carruthers

iSCAM2OM *Reads MLE estimates from iSCAM file structure into an operating model*

Description

A function that uses the file location of a fitted iSCAM model including input files to population the various slots of an operating model parameter estimates. iSCAM2OM relies on several functions written by Chris Grandin (DFO PBS).

Usage

```
iSCAM2OM(
  iSCAMdir,
  nsim = 48,
  proyears = 50,
  mcmc = F,
  Name = NULL,
  Source = "No source provided",
  length_timestep = 1,
  Author = "No author provided"
)
```

Arguments

iSCAMdir	A folder with iSCAM input and output files in it
nsim	The number of simulations to take for parameters with uncertainty (for OM@cpars custom parameters)
proyears	The number of MSE projection years
mcmc	Whether to use mcmc samples to create custom parameters cpars
Name	The name of the operating model
Source	Reference to assessment documentation e.g. a url
length_timestep	How long is a model time step in years (e.g. a quarterly model is 0.25, a monthly model 1/12)
Author	Who did the assessment

Author(s)

T. Carruthers

iSCAMcomps *Combines all iSCAM age composition data across fleets*

Description

iSCAM assessments are often fitted to numerous fleets that have differing age selectivities. iSCAMcomps is a simple way of providing the aggregate catch at age data. It should be noted that this process is important and in a real application would require due diligence (ie peer reviewed data workshop).

Usage

```
iSCAMcomps(replist, Year)
```

Arguments

replist	S3 class object: the output from a read from an iSCAM data folder
Year	Integer vector: the years of the DLMtool data object ie Data@Year

Author(s)

T. Carruthers

iSCAMinds *Combines indices into a single index using linear modelling*

Description

iSCAM assessments often make use of multiple indices of abundance. The DLMtool data object and MPs currently only make use of a single index. combiSCAMinds is a function that creates a single index from many using linear modelling. It is a simple way of providing initial calculations of management recommendations and it should be noted that this process is important and in a real application would require due diligence (ie peer reviewed data workshop).

Usage

```
iSCAMinds(idata, Year, fleeteffect = T)
```

Arguments

idata	List: the indices recorded in a read from an iSCAM data folder, e.g. replist$data$indices
Year	Integer vector: the years of the DLMtool data object ie Data@Year
fleeteffect	Logical: should a fleet effect be added to the linear model?

Author(s)

T. Carruthers

ldim *Dimensions of a hierarchical list object*

Description

Dimensions of a hierarchical list object

Usage

```
ldim(x)
```

Arguments

x	A list

Author(s)

T. Carruthers

load.iscam.files *Reads iSCAM files into a hierarchical R list object*

Description

A function for reading iSCAM input and output files into R

Usage

```
load.iscam.files(model.dir, burnin = 1000, thin = 1, verbose = FALSE)
```

Arguments

model.dir	An iSCAM directory
burnin	The initial mcmc samples to be discarded
thin	The degree of chain thinning 1 in every thin iterations is kept
verbose	Should detailed outputs be provided.

Author(s)

Chris Grandin (DFO PBS)

mahplot — *Plot statistical power of the indicator with increasing time blocks*

Description

Plot statistical power of the indicator with increasing time blocks

Usage

```
mahplot(outlist, res = 6, maxups = 5, MPs)
```

Arguments

outlist	A list object produced by the function PRBcalc
res	Integer, the resolution (time blocking) for the calculation of PPD
maxups	Integer, the maximum number of update time blocks to plot
MPs	Character vector of MP names

Author(s)

T. Carruthers

References

Carruthers and Hordyk 2018

makemov — *Calculates movement matrices from user inputs for fraction in each area (fracs) and probability of staying in areas (prob)*

Description

A function for calculating a movement matrix from user specified unfished stock biomass fraction in each area. Used by simmov to generate movement matrices for a DLMtool operating model.

Usage

```
makemov(fracs = c(0.1, 0.2, 0.3, 0.4), prob = c(0.5, 0.8, 0.9, 0.95))
```

Arguments

fracs	A vector nareas long of fractions of unfished stock biomass in each area
prob	A vector of the probability of individuals staying in each area or a single value for the mean probability of staying among all areas

Author(s)

T. Carruthers

See Also

simmov

`make_MP` *Make a custom management procedure (MP)*

Description

Function operator that combines a function of class `Assess` and a function of class `HCR` to create a management procedure (MP). The resulting function can then be tested in closed-loop simulation via `runMSE`.

Usage

```
make_MP(.Assess, .HCR, diagnostic = c("none", "min", "full"), ...)
```

Arguments

`.Assess`	A function of class `Assess`.
`.HCR`	A function of class `HCR`.
`diagnostic`	A character string describing if any additional diagnostic information from the assessment models will be collected during a call with `runMSE` (`"none"` is the default). `"min"` (minimal) will collect information on convergence and `"full"` will also collect the Assessment object generated by the `.Assess`. This information will be written to the `Misc` slot in the MSE object. See example.
`...`	Additional arguments to be passed to `.Assess` and `.HCR`.

Value

A function of class `MP`.

See Also

HCR_ramp HCR_MSY diagnostic_AM retrospective_AM

Examples

```
# A delay-difference model with a 40-10 control rule
DD_40_10 <- make_MP(DD_TMB, HCR40_10)

# A delay difference model that will produce convergence diagnostics
DD_40_10 <- make_MP(DD_TMB, HCR40_10, diagnostic = "min")

# MP that uses a Delay-Difference which assumes a Ricker stock-recruit function.
DD_Ricker <- make_MP(DD_TMB, HCR_MSY, SR = "Ricker")

## Not run:
myMSE <- DLMtool::runMSE(DLMtool::testOM, MPs = c("FMSYref", "DD_40_10"), PPD = TRUE)

str(myMSE@Misc)
diagnostic_AM(myMSE)

## End(Not run)
```

MMSE-class *Class* 'MMSE'

Description

A Multi Management Strategy Evaluation object that contains information about simulation conditions and performance of MPs for a multi-stock, multi-fleet operating model.

Slots

`Name` Name of the MMSE object. Single value. Character string

`nyears` The number of years for the historical simulation. Single value. Positive integer

`proyears` The number of years for the projections - closed loop simulations. Single value. Positive integer

`nMPs` Number of management procedures simulation tested. Single value. Positive integer.

`MPs` The names of the MPs that were tested. Vector of length nMPs. Character strings.

`MPcond` The MP condition. Character ('bystock': an MP per stock, 'byfleet' and MP per stock and fleet, 'MMP' an MP for all stocks and fleets)

`MPrefs` The names of the MPs applied for each stock (row) and fleet (column). An array.

`nsim` Number of simulations. Single value. Positive integer

`nstocks` Number of stocks. Single value. Positive integer

`nfleets` Number of fleets. Single value. Positive integer

`Snames` Names of the stocks

`Fnames` Names of the fleets (matrix nstocks x nfleets)

`Stocks` The stock operating model objects. List of Stocks

`Fleets` The fleet operating model objects. Hierarchical list, fleets nested in stocks.

`Obss` The fleet specific observation error operating model objects. Hierarchical list, fleets nested in stocks.

`Imps` The fleet soecific implementation error operating model objects. Hierarchical list, fleets nested in stocks.

`OM` A table of sampled parameters of the operating model. Data frame of nsim rows.

`Obs` A table of sampled parameters of the observation model. Data frame of nsim rows.

`B_BMSY` Simulated biomass relative to BMSY over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`F_FMSY` Simulated fishing mortality rate relative to FMSY over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`B` Simulated stock biomass over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`SSB` Simulated spawning stock biomass over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`VB` Simulated vulnerable biomass over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`FM` Simulated fishing mortality rate over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`C` Simulated catches (taken) over the projection. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`TAC` Simulated Total Allowable Catch (prescribed) over the projection (this is NA for input controls). An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`SSB_hist` Simulated historical spawning stock biomass. An array with dimensions: nsim, nages, nyears, nareas. Non-negative real numbers

`CB_hist` Simulated historical catches in weight. An array with dimensions: nsim, nages, nyears, nareas. Non-negative real numbers

`FM_hist` Simulated historical fishing mortality rate. An array with dimensions: nsim, nages, nyears, nareas. Non-negative real numbers

`Effort` Simulated relative fishing effort in the projection years. An array with dimensions: nsim, nMPs, proyears. Non-negative real numbers

`PAA` Population at age in last projection year. An array with dimensions: nsim, nMPs, nages. Non-negative real numbers

`CAA` Catch at age in last projection year. An array with dimensions: nsim, nMPs, nages. Non-negative real numbers

`CAL` Catch at length in last projection year. An array with dimensions: nsim, nMPs, nCALbins. Non-negative real numbers

`CALbins` Mid-points of the catch-at-length bins. Vector of length nCALbins. Positive real numbers.

`MSY_P` Array of projected MSY by year with dimensions: nsim, nstock, nMP, proyears.

`FMSY_P` Array of projected FMSY by year with dimensions: nsim, nstock, nMP, proyears.

`SSBMSY_P` Array of projected Spawning Stock Biomass at MSY by year with dimensions: nsim, nstock, nMP, proyears.

`Misc` Miscellanenous output such as posterior predictive data

Objects from the Class

Objects can be created by calls of the form `new('MMSE',Name,nyears,proyears,nMPs,MPs,nsim,OMtable,Obs,B_BMSYa`

Author(s)

T. Carruthers

MOM-class *Class* 'MOM'

Description

An object containing all the parameters needed to control a multi-stock, multi-fleet MSE which can be build from component Stock, Fleet, Obs, and Imp objects.

Details

Almost all of these inputs are a vector of length 2 which describes the upper and lower bounds of a uniform distribution from which to sample the parameter.

Slots

`Name` Name of the operating model

`Agency` Name of the agency responsible for the management of the fishery. Character string

`Region` Name of the general geographic region of the fishery. Character string

`Sponsor` Name of the organization who sponsored the OM. Character string

`Latitude` Latitude (decimal degrees). Negative values represent the South of the Equator. Numeric. Single value

`Longitude` Longitude (decimal degrees). Negative values represent the West of the Prime Meridian. Numeric. Single value

`nsim` The number of simulations

`proyears` The number of projected years

`interval` The assessment interval - how often would you like to update the management system?

`pstar` The percentile of the sample of the management recommendation for each method

`maxF` Maximum instantaneous fishing mortality rate that may be simulated for any given age class

`reps` Number of samples of the management recommendation for each method. Note that when this is set to 1, the mean value of the data inputs is used.

`cpars` A hierarcical list nstock then nfleet long of custom parameters. Time series are a matrix nsim rows by nyears columns. Single parameters are a vector nsim long

`seed` A random seed to ensure users can reproduce results exactly

`Source` A reference to a website or article from which parameters were taken to define the operating model

`Stocks` List of stock objects

`Fleets` List of Fleet objects

`Obs` Hierarchical List of Observation model objects Level 1 is stock, level 2 is fleet

`Imps` Hierarchical List of Implementation model objects Level 1 is stock, level 2 is fleet

`CatchFrac` A list nstock long, of matrices nsim x nfleet representing the fraction of current catches of the various fleets to each stock (each matrix is nsim by nfleet long and rows sum to 1 for each stock)

`Allocation` A list nstock long, of matrices nsim x nfleet representing the fraction of future TACs of the various fleets to each stock (each matrix is nsim by nfleet long and rows sum to 1 for each stock).

`Efactor` A list nstock long, of current effort factors by fleet (default is 1 - same as current effort)

`Complexes` A list of stock complexes. Each position is a vector of stock numbers (as they appear in StockPars) for which data should be aggregated and TAC recommendations split among stocks according to vulnerable biomass

`SexPars` A list of slots that control sex-specific dynamics

`Rel` A list of biological / ecological relationships among stocks over-ridden if an MP of class 'MP_F" is supplied that is a multi-fleet MP.

Objects from the Class

Objects can be created by calls of the form `new('MOM',Stock_list,Fleet_list,Obs_list,Imp_list)`.

Author(s)

T. Carruthers and A. Hordyk

MPCalcsNAs *Fill any NAs arising from MPCalcs (hermaphroditism mode)*

Description

Fill any NAs arising from MPCalcs (hermaphroditism mode)

Usage

```
MPCalcsNAs(MPCalcs)
```

Arguments

`MPCalcs`	A list of arrays arising fromt the DLMtool function CalcMPDynamics()

Author(s)

T. Carruthers

multiData *Combine data among fleets*

Description

Catches, CAA, CAL are summed. LFC and LFS are weighted averages. ML, Lc and Lbar are recalculated from summed CAL. All other observations are for fleet 1 (indicative)

Usage

```
multiData(MSElist, StockPars, p, mm, nf)
```

Arguments

MSElist	A hierarcical list of data objects stock then fleet then MP
StockPars	A list of stock parameters
p	Integer the Stock number
mm	Integer the MP number
nf	The number of fleets

Author(s)

T. Carruthers

multiDataS *Combine data among stocks*

Description

Catches, CAA, CAL are summed. LFC and LFS are weighted averages. ML, Lc and Lbar are recalculated from summed CAL. All other observations are for fleet 1 (indicative)

Usage

```
multiDataS(MSElist, StockPars, np, mm, nf, realVB)
```

Arguments

MSElist	A hierarcical list of data objects stock then fleet then MP
StockPars	A list of stock parameters
np	The number of stocks
mm	Integer the MP number
nf	The number of fleets
realVB	A matrix of real vulnerable biomass [nsim,year,np]

Author(s)

T. Carruthers

`multidebug` *A basic comparison of runMSE output (MSE) and multiMSE (MMSE)*

Description

A basic comparison of runMSE output (MSE) and multiMSE (MMSE)

Usage

```
multidebug(MSEsingle, MSEmulti, p = 1, f = 1, MPno = 1, maxsims = 4)
```

Arguments

`MSEsingle`	An object of class MSE arising from a run of runMSE(OM, ...)
`MSEmulti`	An object of class MMSE arising from a run of multiMSE(MOM, ...)
`p`	Integer. The stock number from the MSEmulti object (to be plotted)
`f`	Integer. The fleet number from the MSEmulti object (to be plotted)
`MPno`	Integer. The MP number from the MSEmulti and MSEsingle object (to be plotted)
`maxsims`	Integer. The maximum number of simulations to plot.

Author(s)

T.Carruthers

`multiMSE` *Run a multi-fleet multi-stock Management Strategy Evaluation*

Description

A function that runs a Management Strategy Evaluation (closed-loop simulation) for a specified operating model

Usage

```
multiMSE(
  MOM,
  MPs = list(c("AvC", "DCAC"), c("FMSYref", "curE")),
  CheckMPs = FALSE,
  timelimit = 1,
  Hist = FALSE,
  ntrials = 50,
  fracD = 0.05,
  CalcBlow = FALSE,
  HZN = 2,
  Bfrac = 0.5,
  AnnualMSY = TRUE,
  silent = FALSE,
  PPD = FALSE,
  parallel = FALSE,
  save_name = NULL,
  checks = FALSE,
  control = NULL
)
```

Arguments

MOM	A multi-fleet multi-stock operating model (class 'MOM')
MPs	A matrix of methods (nstock x nfleet) (character string) of class MP
CheckMPs	Logical to indicate if Can function should be used to check if MPs can be run.
timelimit	Maximum time taken for a method to carry out 10 reps (methods are ignored that take longer)
Hist	Should model stop after historical simulations? Returns a list containing all historical data
ntrials	Maximum of times depletion and recruitment deviations are resampled to optimize for depletion. After this the model stops if more than percent of simulations are not close to the required depletion
fracD	Maximum allowed proportion of simulations where depletion is not close to sampled depletion from OM before model stops with error
CalcBlow	Should low biomass be calculated where this is the spawning biomass at which it takes HZN mean generation times of zero fishing to reach Bfrac fraction of SSBMSY
HZN	The number of mean generation times required to reach Bfrac SSBMSY in the Blow calculation
Bfrac	The target fraction of SSBMSY for calculating Blow
AnnualMSY	Logical. Should MSY statistics be calculated for each projection year? May differ from MSY statistics from last historical year if there are changes in productivity
silent	Should messages be printed out to the console?

PPD	Logical. Should posterior predicted data be included in the MSE object Misc slot?
parallel	Logical. Should the MSE be run using parallel processing?
save_name	Character. Optional name to save parallel MSE list
checks	Logical. Run tests?
control	control options for testing and debugging

Value

A hierarchical list (by stock then fleet) of objects of class MSE

Author(s)

T. Carruthers and A. Hordyk

NIL *Item in list: get the list values from a list of lists*

Description

Create of vector of values that correspond with a slot in a list of objects

Usage

```
NIL(listy, namey, lev1 = T)
```

Arguments

listy	A list of objects
namey	A character vector representing the list item's name
lev1	Logical, should NIL default to the first level of the list?

Author(s)

T. Carruthers

`plot.Assessment` *Plot Assessment object*

Description

Produces HTML file (via markdown) figures of parameter estimates and output from an Assessment object.

Usage

```
## S4 method for signature 'Assessment,missing'
plot(
  x,
  filename = paste0("report_", x@Model),
  dir = tempdir(),
  ret_yr = 0L,
  open_file = TRUE,
  quiet = TRUE,
  ...
)

## S4 method for signature 'Assessment,retro'
plot(
  x,
  y,
  filename = paste0("report_", x@Model),
  dir = tempdir(),
  open_file = TRUE,
  quiet = TRUE,
  ...
)
```

Arguments

`x`	An object of class Assessment.
`filename`	Character string for the name of the markdown and HTML files.
`dir`	The directory in which the markdown and HTML files will be saved.
`ret_yr`	If greater than zero, then a retrospective analysis will be performed and results will be reported. The integer here corresponds to the number of peels (the maximum number of terminal years for which the data are removed).
`open_file`	Logical, whether the HTML document is opened after it is rendered.
`quiet`	Logical, whether to silence the markdown rendering function.
`...`	Other arguments to pass to render.
`y`	An object of class retro.

Value

Returns invisibly the output from render.

See Also

retrospective

Examples

```
output <- DD_TMB(Data = Simulation_1)

## Not run:
plot(output)

## End(Not run)
```

---

plot.MMSE *Standard plot for an object of class MMSE (multi MSE)*

Description

Plot the projected biomass, fishing, mortality rate and yield for all stocks and MPs

Usage

```
## S4 method for signature 'MMSE,missing'
plot(
  x,
  maxcol = 6,
  qcol = rgb(0.4, 0.8, 0.95),
  lcol = "dodgerblue4",
  quants = c(0.05, 0.25, 0.75, 0.95),
  curyr = 2018,
  addline = FALSE
)
```

Arguments

x	Object of class MMSE. A Multi-OM object created by multiMSE(MOM,...)
maxcol	Integer. The maximum number of columns (MPs) to be plotted in each plot
qcol	Character, color. The color of the inner percentile range
lcol	Character, color. The color of the outer percentile range.
quants	Numeric vector. The percentiles that are plotted (LB2,LB1,UB1,UB2). LB2 and UB2 are the outer percentiles, LB1 and UB1 are the inner percentiles.
curyr	Integer. The current year from which projections start.
addline	Logical. Should two individual simulations be added to the percentile plots?

Author(s)

T.Carruthers

`plot.MOM` *Standard plot for an object of class MOM*

Description

Plot the stocks, fleets, catch fractions and relationships in multi operating model object

Usage

```
## S4 method for signature 'MOM,missing'
plot(x, silent = TRUE, maxsims = 6)
```

Arguments

`x`	Object of class MOM. A Multi-OM object created by `new('MOM',...)`
`silent`	Logical. Do you wish to see print outs / warnings?
`maxsims`	Integer. What are the maximum number of individual simulations you wish to plot?

Author(s)

T.Carruthers

`plot.prof` *Plot profile object*

Description

Generates a profile plot generated by profile. If a two-parameter profile is performed, then a contour plot of the likelihood surface is returned.

Usage

```
## S4 method for signature 'prof,missing'
plot(x, contour_levels = 20, ...)
```

Arguments

`x`	An object of class prof returned by profile.
`contour_levels`	Integer, passed to `nlevels` argument of contour.
`...`	Miscellaneous. Not used.

Author(s)

Q. Huynh

`plot.retro` *Methods for retro object*

Description

plot and summary functions for retro object.

Usage

```
## S4 method for signature 'retro,missing'
plot(x, color = NULL)

## S4 method for signature 'retro'
summary(object)
```

Arguments

`x`	An object of class retro.
`color`	An optional character vector of colors for plotting.
`object`	An object of class retro.

Author(s)

Q. Huynh

Examples

```
res <- SCA(Data = DLMtool::Red_snapper)
ret <- retrospective(res)

summary(ret)

## Not run:
plot(ret)

## End(Not run)
```

plot.SRA *Plot SRA scope output*

Description

Produces HTML file (via markdown) figures of parameter estimates and output from an Assessment object. Plots histograms of operating model parameters that are updated by the SRA scoping function, as well as diagnostic plots for the fits to the SRA for each simulation. `compare_SRA` plots a short report that compares output from multiple SRA objects, assuming the same model structure but different data weightings, data omissions, etc.

Usage

```
## S4 method for signature 'SRA,missing'
plot(
  x,
  compare = TRUE,
  filename = "SRA_scope",
  dir = tempdir(),
  sims = 1:x@OM@nsim,
  Year = NULL,
  f_name = NULL,
  s_name = NULL,
  MSY_ref = c(0.5, 1),
  bubble_adj = 10,
  scenario = list(),
  title = NULL,
  open_file = TRUE,
  quiet = TRUE,
  render_args,
  ...
)

compare_SRA(
  ...,
  compare = TRUE,
  filename = "compare_SRA",
  dir = tempdir(),
  Year = NULL,
  f_name = NULL,
  s_name = NULL,
  MSY_ref = c(0.5, 1),
  bubble_adj = 10,
  scenario = list(),
  title = NULL,
  open_file = TRUE,
  quiet = TRUE,
```

```
  render_args
)
```

Arguments

x	An object of class SRA (output from SRA_scope).
compare	Logical, if TRUE, the function will run runMSE to compare the historical period of the operating model and the SRA model output.
filename	Character string for the name of the markdown and HTML files.
dir	The directory in which the markdown and HTML files will be saved.
sims	A logical vector of length x@OM@nsim or a numeric vector indicating which simulations to keep.
Year	Optional, a vector of years for the historical period for plotting.
f_name	Character vector for fleet names.
s_name	Character vector for survey names.
MSY_ref	A numeric vector for reference horizontal lines for B/BMSY plots.
bubble_adj	A number to adjust the size of bubble plots (for residuals of age and length comps).
scenario	Optional, a named list to label each simulation in the SRA for plotting, e.g.: list(names = c("low M","high M"),col = c("blue","red")).
title	Optional character string for an alternative title for the markdown report.
open_file	Logical, whether the HTML document is opened after it is rendered.
quiet	Logical, whether to silence the markdown rendering function.
render_args	A list of other arguments to pass to render.
...	For compare_SRA, multiple SRA objects for comparison.

Value

Returns invisibly the output from render.

See Also

SRA SRA_scope

plotmulti *A basic SSB plot for debugging runMSE output*

Description

A basic SSB plot for debugging runMSE output

Usage

```
plotmulti(MSEmulti, maxsim = 8)
```

Arguments

MSEmulti	An object of class MMSE arising from a run of multiMSE(MOM, ...)
maxsim	Integer. The number of simulations to plot

Author(s)

T.Carruthers

plotquant *A fairly tidy time-series quantile plot*

Description

A fairly tidy time-series quantile plot

Usage

```
plotquant(
  x,
  p = c(0.05, 0.25, 0.75, 0.95),
  yrs,
  qcol,
  lcol,
  addline = T,
  ablines = NA
)
```

Arguments

x	Matrix. A time series quantity [simulation, year]
p	Numeric vector. The percentiles that are plotted (LB2,LB1,UB1,UB2). LB2 and UB2 are the outer percentiles, LB1 and UB1 are the inner percentiles.
yrs	Numeric vector. The years corresponding to the indexing of x
qcol	Character, color. The color of the inner percentile range
lcol	Character, color. The color of the outer percentile range.
addline	Logical. Should two individual simulations be added to the percentile plots?
ablines	Numeric vector. Horizontal lines to be added to the plot.

Author(s)

T.Carruthers

---

plotRel *Plot a relationship between stocks*

Description

Plot a relationship between stocks

Usage

```
plotRel(Stocks, Rel, Relno, Snams, leg = F, extras = 0)
```

Arguments

Stocks	A list of stock objects (MOM@Stocks)
Rel	A list of inter-stock MICE relationships (MOM@Rel)
Relno	Integer. The relationship you wish to plot
Snams	A vector of stock names
leg	Logical. Do you want to plot a legend?
extras	Integer. The number of blank plots to create at the end.

Author(s)

T.Carruthers

plot_betavar *Plots a beta variable*

Description

Plots the probability distribution function of a beta variable from the mean and standard deviation in either transformed (logit) or untransformed space.

Usage

```
plot_betavar(m, sd, label = NULL, is_logit = FALSE, color = "black")
```

Arguments

m	A vector of means of the distribution.
sd	A vector of standard deviations of the distribution.
label	Name of the variable to be used as x-axis label.
is_logit	Logical that indicates whether the means and standard deviations are in transformed (logit) or untransformed space.
color	A vector of colors.

Value

A plot of the probability distribution function. Vertical dotted line indicates mean of distribution. This function can plot multiple curves when multiple means and standard deviations are provided.

Author(s)

Q. Huynh

See Also

plot_lognormalvar plot_steepness

Examples

```
mu <- 0.5
stddev <- 0.1
plot_betavar(mu, stddev) # mean of plot should be 0.5

#logit parameters
mu <- 0
stddev <- 0.1
plot_betavar(mu, stddev, is_logit = TRUE) # mean of plot should be 0.5
```

plot_composition *Plot composition data*

Description

Plots annual length or age composition data.

Usage

```
plot_composition(
  Year = 1:nrow(obs),
  obs,
  fit = NULL,
  plot_type = c("annual", "bubble_data", "bubble_residuals", "mean"),
  N = rowSums(obs),
  CAL_bins = NULL,
  ages = NULL,
  ind = 1:nrow(obs),
  annual_ylab = "Frequency",
  annual_yscale = c("proportions", "raw"),
  bubble_adj = 5,
  fit_linewidth = 3,
  fit_color = "red"
)
```

Arguments

Year	A vector of years.
obs	A matrix of either length or age composition data. For lengths, rows and columns should index years and length bin, respectively. For ages, rows and columns should index years and age, respectively.
fit	A matrix of predicted length or age composition from an assessment model. Same dimensions as obs.
plot_type	Indicates which plots to create. Options include annual distributions, bubble plot of the data, and bubble plot of the residuals, and annual means.
N	Annual sample sizes. Vector of length nrow(obs).
CAL_bins	A vector of lengths corresponding to the columns in obs. and fit. Ignored for age data.
ages	An optional vector of ages corresponding to the columns in obs.
ind	A numeric vector for plotting a subset of rows (which indexes year) of obs and fit.
annual_ylab	Character string for y-axis label when plot_type = "annual".
annual_yscale	For annual composition plots (plot_type = "annual"), whether the raw values ("raw") or frequencies ("proportions") are plotted.

bubble_adj	Numeric, for adjusting the relative size of bubbles in bubble plots (larger number = larger bubbles).
fit_linewidth	Argument lwd for fitted line.
fit_color	Color of fitted line.

Value

Plots depending on plot_type.

Author(s)

Q. Huynh

Examples

```
data(Red_snapper)
plot_composition(obs = Red_snapper@CAA[1, , ], plot_type = "annual")
plot_composition(obs = Red_snapper@CAA[1, , ], plot_type = "bubble_data")

plot_composition(obs = Red_snapper@CAL[1, , ], plot_type = "annual", Red_snapper@CAL_bins[1:43])
plot_composition(obs = Red_snapper@CAL[1, , ], plot_type = "bubble_data",
CAL_bins = Red_snapper@CAL_bins[1:43])
```

---

plot_crosscorr *Produce a cross-correlation plot of the derived data arising from getinds(MSE_object)*

Description

Produce a cross-correlation plot of the derived data arising from getinds(MSE_object)

Usage

```
plot_crosscorr(
  indPPD,
  indData,
  pp = 1,
  dnam = c("CS", "CV", "CM", "IS", "MLS"),
  res = 1
)
```

Arguments

indPPD	A 3D array of results arising from running getind on an MSE of the Null operating model (type of data/stat (e.g. mean catches),time period (chunk), simulation)
indData	A 3D array of results arising from running getind on an MSE of the Alternative operating model (type of data/stat (e.g. mean catches),time period (chunk), simulation)
pp	Positive integer, the number of time chunks (blocks of years normally, second dimension of indPPD and indData) to produce the plot for.
dnam	A character vector of names of the data for plotting purposes (as long as dimension 1 of indPPD and indData).
res	The size of the temporal blocking that greated indPPD and indData - this is just used for labelling purposes

Value

A cross-correlation plot (ndata-1) x (ndata-1)

Author(s)

T. Carruthers

References

Carruthers and Hordyk 2018

---

plot_lognormalvar *Plots a lognormal variable*

Description

Plots the probability distribution function of a lognormal variable from the mean and standard deviation in either transformed (normal) or untransformed space.

Usage

```
plot_lognormalvar(m, sd, label = NULL, logtransform = FALSE, color = "black")
```

Arguments

m	A vector of means of the distribution.
sd	A vector of standard deviations of the distribution.
label	Name of the variable to be used as x-axis label.
logtransform	Indicates whether the mean and standard deviation are in transformed (normal) or untransformed space.
color	A vector of colors.

Value

A plot of the probability distribution function. Vertical dotted line indicates mean of distribution. This function can plot multiple curves when multiple means and standard deviations are provided.

Author(s)

Q. Huynh

See Also

plot_betavar plot_steepness

Examples

```
mu <- 0.5
stddev <- 0.1
plot_lognormalvar(mu, stddev) # mean of plot should be 0.5

#logtransformed parameters
mu <- 0
stddev <- 0.1
plot_lognormalvar(mu, stddev, logtransform = TRUE) # mean of plot should be 1
```

---

plot_residuals *Plot residuals*

Description

Plots figure of residuals (or any time series with predicted mean of zero).

Usage

```
plot_residuals(
  Year,
  res,
  res_sd = NULL,
  res_sd_CI = 0.95,
  res_upper = NULL,
  res_lower = NULL,
  res_ind_blue = NULL,
  draw_zero = TRUE,
  zero_linetype = 2,
  label = "Residual"
)
```

Arguments

Year	A vector of years for the data.
res	A vector of residuals.
res_sd	A vector of year specific standard deviation for res.
res_sd_CI	The confidence interval for the error bars based for res_sd.
res_upper	A vector of year-specific upper bounds for the error bars of the residual (in lieu of argument res_CV).
res_lower	A vector of year-specific lower bounds for the error bars of the residual (in lieu of argument res_CV).
res_ind_blue	Indices of obs for which the plotted residuals and error bars will be blue.
draw_zero	Indicates whether a horizontal line should be drawn at zero.
zero_linetype	Passes argument lty (e.g. solid line = 1, dotted = 2) to draw_zero.
label	Character string that describes the data to label the y-axis.

Author(s)

Q. Huynh

See Also

plot_timeseries

---

plot_SR *Plot stock-recruitment function*

Description

Plot stock-recruitment (with recruitment deviations if estimated).

Usage

```
plot_SR(
  Spawners,
  expectedR,
  R0 = NULL,
  S0 = NULL,
  rec_dev = NULL,
  trajectory = FALSE,
  y_zoom = NULL,
  ylab = "Recruitment"
)
```

Arguments

Spawners	A vector of the number of the spawners (x-axis).
expectedR	A vector of the expected recruitment (from the stock-recruit function) corresponding to values of Spawners.
R0	Virgin recruitment.
S0	Virgin spawners.
rec_dev	If recruitment deviations are estimated, a vector of estimated recruitment (in normal space) corresponding to values of Spawners.
trajectory	Indicates whether arrows will be drawn showing the trajectory of spawners and recruitment deviations over time.
y_zoom	If recruitment deviations are plotted, the y-axis limit relative to maximum expected recruitment expectedR. If NULL, all recruitments are plotted.
ylab	Character string for label on y-axis.

Value

A stock-recruit plot

Author(s)

Q. Huynh

---

plot_steepness *Plots probability distribution function of stock-recruit steepness*

Description

Plots the probability distribution function of steepness from the mean and standard deviation.

Usage

```
plot_steepness(
  m,
  sd,
  is_transform = FALSE,
  SR = c("BH", "Ricker"),
  color = "black"
)
```

Arguments

m	The mean of the distribution (vectorized).
sd	The standard deviation of the distribution (vectorized).
is_transform	Logical, whether the mean and standard deviation are in normal space (FALSE) or transformed space.
SR	The stock recruitment relationship (determines the range and, if relevant, transformation of steepness).
color	A vector of colors.

Value

A plot of the probability distribution function. Vertical dotted line indicates mean of distribution.

Note

The function samples from a beta distribution with parameters alpha and beta that are converted from the mean and standard deviation. Then, the distribution is transformed from 0 - 1 to 0.2 - 1.

Author(s)

Q. Huynh

See Also

plot_lognormalvar plot_betavar

Examples

```
mu <- DLMtool::Simulation_1@steep
stddev <- DLMtool::Simulation_1@steep * DLMtool::Simulation_1@CV_steep
plot_steepness(mu, stddev)
```

---

plot_timeseries *Plot time series of data*

Description

Plot time series of observed (with lognormally-distributed error bars) vs. predicted data.

Usage

```
plot_timeseries(
  Year,
  obs,
  fit = NULL,
  obs_CV = NULL,
  obs_CV_CI = 0.95,
  obs_upper = NULL,
  obs_lower = NULL,
  obs_ind_blue = NULL,
  fit_linewidth = 3,
  fit_color = "red",
  label = "Observed data"
)
```

Arguments

Year	A vector of years for the data.
obs	A vector of observed data.
fit	A vector of predicted data (e.g., from an assessment model).
obs_CV	A vector of year-specific coefficient of variation in the observed data.
obs_CV_CI	The confidence interval for the error bars based for obs_CV.
obs_upper	A vector of year-specific upper bounds for the error bars of the observed data (in lieu of argument obs_CV).
obs_lower	A vector of year-specific lower bounds for the error bars of the observed data (in lieu of argument obs_CV).
obs_ind_blue	Indices of obs for which the plotted points and error bars will be blue.
fit_linewidth	Argument lwd for fitted line.
fit_color	Color of fitted line.
label	Character string that describes the data to label the y-axis.

Author(s)

Q. Huynh

See Also

plot_residuals

Examples

```
data(Red_snapper)
plot_timeseries(Red_snapper@Year, Red_snapper@Cat[1, ],
obs_CV = Red_snapper@CV_Cat, label = "Catch")
```

PRBcalc *Calculate mahalanobis distance (null and alternative MSEs) and statistical power for all MPs in an MSE*

Description

Calculate mahalanobis distance (null and alternative MSEs) and statistical power for all MPs in an MSE

Usage

```
PRBcalc(
  MSE_null,
  MSE_alt,
  tsd = c("Cat", "Cat", "Cat", "Ind", "ML"),
  stat = c("slp", "AAV", "mu", "slp", "slp"),
  dnam = c("C_S", "C_V", "C_M", "I_S", "ML_S"),
  res = 6,
  alpha = 0.05,
  plotCC = FALSE,
  removedat = FALSE,
  removethresh = 0.025
)
```

Arguments

MSE_null	An object of class MSE representing the null hypothesis
MSE_alt	An object of class MSE representing the alternative hypothesis
tsd	Character string of data types: Cat = catch, Ind = relative abundance index, ML = mean length in catches
stat	Character string defining the quantity to be calculated for each data type, slp = slope(log(x)), AAV = average annual variability, mu = mean(log(x))
dnam	Character string of names for the quantities calculated
res	Integer, the resolution (time blocking) for the calculation of PPD
alpha	Probability of incorrectly rejecting the null operating model when it is valid
plotCC	Logical, should the PPD cross correlations be plotted?
removedat	Logical, should data not contributing to the mahalanobis distance be removed?
removethresh	Positive fraction: the cumulative percentage of removed data (removedat=TRUE) that contribute to the mahalanobis distance

Value

A list object with two hierarchies of indexing, first by MP, second has two positions as described in Probs: (1) mahalanobis distance, (2) a matrix of type 1 error (first row) and statistical power (second row), by time block.

Author(s)

T. Carruthers

References

Carruthers, T.R, and Hordyk, A.R. In press. Using management strategy evaluation to establish indicators of changing fisheries. Canadian Journal of Fisheries and Aquatic Science.

---

prelim_AM *Preliminary Assessments in MSE*

Description

Evaluates the likely performance of Assessment models in the operating model. This function will apply the assessment model for Data generated during the historical period of the MSE, and report the convergence rate for the model and total time elapsed in running the assessments.

Usage

```
prelim_AM(x, Assess, ncpus = NULL, ...)
```

Arguments

x	Either a Hist, Data or OM object.
Assess	An Assess function of class Assess.
ncpus	Numeric, the number of CPUs to run the Assessment model (will run in parallel if greater than 1).
...	Arguments to be passed to Assess, e.g., model configurations.

Value

Returns invisibly a list of Assessment objects of length OM@nsim. Messages via console.

Author(s)

Q. Huynh

Examples

```
## Not run:
prelim_AM(DLMtool::testOM, DD_TMB)

## End(Not run)
```

`Probs` *Calculates mahalanobis distance and rejection of the Null operating model*

Description

Calculates mahalanobis distance and rejection of the Null operating model, used by wrapping function PRBcalc.

Usage

```
Probs(indPPD, indData, alpha = 0.05, removedat = FALSE, removethresh = 0.05)
```

Arguments

`indPPD`	A 3D array of results arising from running getind on an MSE of the Null operating model (type of data/stat (e.g. mean catches),time period (chunk), simulation)
`indData`	A 3D array of results arising from running getind on an MSE of the Alternative operating model (type of data/stat (e.g. mean catches),time period (chunk), simulation)
`alpha`	Positive fraction: rate of type I error, alpha
`removedat`	Logical, should data not contributing to the mahalanobis distance be removed?
`removethresh`	Positive fraction: the cumulative percentage of removed data (removedat=TRUE) that contribute to the mahalanobis distance

Value

A list object. Position 1 is an array of the mahalanobis distances. Dimension 1 is length 2 for the Null OM (indPPD) and the alternative OM (indData). Dimension 2 is the time block (same length as indPPD dim 2). Dimension 3 is the simulation number (same length at indPPD dim 3.), Position 2 is a matrix (2 rows, ntimeblock columns) which is (row 1) alpha: the rate of false positives, and row 2 the power (1-beta) the rate of true positives

Author(s)

T. Carruthers

References

Carruthers and Hordyk 2018

prof-class *Class*-prof

Description

An S4 class that contains output from profile.

Slots

`Model` Name of the assessment model.

`Name` Name of Data object.

`Par` Character vector of parameters that were profiled.

`MLE` Numeric vector of the estimated values of the parameters (corresponding to `Par`) from the assessment.

`grid` A data.frame of the change in negative log-likelihood (`nll`) based on the profile of the parameters.

Author(s)

Q. Huynh

See Also

plot.prof profile

profile *Profile likelihood of assessment models*

Description

Profile the likelihood for parameters of assessment models.

Usage

```
profile(fitted, ...)

## S4 method for signature 'Assessment'
profile(fitted, figure = TRUE, ...)

profile_likelihood(Assessment, figure = TRUE, ...)
```

Arguments

`fitted, Assessment`	An object of class Assessment.
`...`	A sequence of values of the parameter(s) for the profile. See details and example below. See details for name of arguments to be passed on.
`figure`	Logical, indicates whether a figure will be plotted.

Details

As of version 1.2, `profile_likelihood` is deprecated in favor of `profile`.

For the following assessment models, possible sequence of values for profiling are:

- DD_TMB and DD_SS: `R0` and `h`
- SP and SP_SS: `FMSY` and `MSY`
- DD and cDD_SS: `R0` and `h`
- SCA and SCA_Pope: `R0` and `h`
- SCA2: `meanR`
- VPA: `F_term`

Value

An object of class prof that contains a data frame of negative log-likelihood values from the profile and, optionally, a figure of the likelihood surface.

Author(s)

Q. Huynh

Examples

```
output <- DD_TMB(Data = DLMtool::Red_snapper)
pro <- profile(output, R0 = seq(0.75, 1.25, 0.025), h = seq(0.9, 0.99, 0.01))
pro <- profile(output, R0 = seq(0.75, 1.25, 0.025)) # Profile R0 only

# Ensure your grid is of proper resolution. A grid that is too coarse
# will likely distort the shape of the likelihood surface.
```

project-class *Class*-project

Description

An S4 class for the output from projection.

Slots

`Model` Name of the assessment model.

`Name` Name of Data object.

`FMort` A matrix of fishing mortality over `p_sim` rows and `p_years` columns.

`B` An matrix of biomass with `p_sim` rows and `p_years` columns.

`SSB` A matrix of spawning biomass with `p_sim` rows and `p_years` columns.

`VB` A matrix of vulnerable biomass with `p_sim` rows and `p_years` columns.

`R` A matrix of recruitment over `p_sim` rows and `p_years` columns.

`N` A matrix of abundance over `p_sim` rows and `p_years` columns.

`Catch` A matrix of observed catch over `p_sim` rows and `p_years` columns.

`Index` A matrix of observed index over `p_sim` rows and `p_years` columns.

`C_at_age` An array for catch-at-age with dimension `c(p_sim,p_years,maxage)`.

Author(s)

Q. Huynh

See Also

projection

projection *Projections for assessment models*

Description

This function takes an assessment model and runs a stochastic projection based on future F or catch.

Usage

```
projection(
  Assessment,
  constrain = c("F", "Catch"),
  FMort = NULL,
  Catch = NULL,
  p_years = 50,
  p_sim = 200,
  obs_error = NULL,
  process_error = NULL,
  max_F = 3,
  seed = 499
)
```

Arguments

Assessment	An object of class Assessment.
constrain	Whether to project on future F or catch. By default, projects on F.
FMort	The projection F, either of length 1 for constant F for the entirety of the projection or length p_years.
Catch	The projection catch, either of length 1 for constant catch for the entirety of the projection or length p_years.
p_years	Integer for the number of projection years.
p_sim	Integer for the number of simulations for the projection.
obs_error	Vector of length two for standard deviation of error to be added to the index and catch, respectively. If NULL, uses values from assessment model.
process_error	Numeric, standard deviation for process error (e.g., recruitment or biomass deviates). If NULL, uses values from assessment model.
max_F	The maximum allowable F if the projection is constrained on catch.
seed	An integer to set the seed for the sampling observation and process error deviates.

Examples

```
myAssess <- SCA(Data = SimulatedData)
do_projection <- projection(myAssess, FMort = myAssess@FMSY)
```

read.control.file *Reads iSCAM control file*

Description

A function for returning the results of the iscam control file

Usage

```
read.control.file(
  file = NULL,
  num.gears = NULL,
  num.age.gears = NULL,
  verbose = FALSE
)
```

Arguments

file	File location
num.gears	The number of gears
num.age.gears	The number age-gears
verbose	should detailed results be printed to console

Author(s)

Chris Grandin (DFO PBS)

read.data.file *Reads iSCAM dat file*

Description

A function for returning the results of the .dat iscam file

Usage

```
read.data.file(file = NULL, verbose = FALSE)
```

Arguments

file	File location
verbose	should detailed results be printed to console

Author(s)

Chris Grandin (DFO PBS)

read.mcmc *Reads iSCAM mcmc output files*

Description

A function for returning the results of the iscam mcmc files

Usage

```
read.mcmc(model.dir = NULL, verbose = TRUE)
```

Arguments

model.dir	Folder name
verbose	should detailed results be printed to console

Author(s)

Chris Grandin (DFO PBS)

read.par.file *Reads iSCAM parameter file*

Description

A function for returning the results of the iscam .par file

Usage

```
read.par.file(file = NULL, verbose = FALSE)
```

Arguments

file	File location
verbose	should detailed results be printed to console

Author(s)

Chris Grandin (DFO PBS)

read.projection.file *Reads iSCAM projection file*

Description

A function for returning the results of the iscam projection file

Usage

```
read.projection.file(file = NULL, verbose = FALSE)
```

Arguments

file	File location
verbose	should detailed results be printed to console

Author(s)

Chris Grandin (DFO PBS)

read.report.file *Reads iSCAM Rep file*

Description

A function for returning the results of the .rep iscam file

Usage

```
read.report.file(fn)
```

Arguments

fn	File location

Author(s)

Chris Grandin (DFO PBS)

retro-class *Class*-retro

Description

An S4 class that contains output from retrospective.

Slots

`Model` Name of the assessment model.

`Name` Name of Data object.

`TS_var` Character vector of time series variables, e.g. recruitment, biomass, from the assessment.

`TS` An array of time series assessment output of dimension, indexed by: peel (the number of terminal years removed from the base assessment), years, and variables (corresponding to `TS_var`).

`Est_var` Character vector of estimated paramters, e.g. R0, steeppness, in the assessment.

`Est` An array for estimated parameters of dimension, indexed by: peel, variables (corresponding to `Est_var`), and value (length 2 for estimate and standard error).

Author(s)

Q. Huynh

See Also

plot.retro summary.retro plot.Assessment

retrospective *Retrospective analysis of assessment models*

Description

Perform a retrospective analysis, successive removals of most recent years of data to evaluate resulting parameter estimates.

Usage

```
retrospective(x, ...)

## S4 method for signature 'Assessment'
retrospective(x, nyr = 5, figure = TRUE)
```

Arguments

x	An S4 object of class Assessment.
...	More arguments.
nyr	The maximum number of years to remove for the retrospective analysis.
figure	Indicates whether plots will be drawn.

Value

A list with an array of model output and of model estimates from the retrospective analysis.

Figures showing the time series of biomass and exploitation and parameter estimates with successive number of years removed. For a variety of time series output (SSB, recruitment, etc.) and estimates (R0, steepness, etc.), also returns a matrix of Mohn's rho (Mohn 1999).

Author(s)

Q. Huynh

References

Mohn, R. 1999. The retrospective problem in sequential population analysis: an investigation using cod fishery and simulated data. ICES Journal of Marine Science 56:473-488.

Examples

```
output <- DD_TMB(Data = DLMtool::Red_snapper)
get_retro <- retrospective(output, nyr = 5, figure = FALSE)
```

---

retrospective_AM *retrospective_AM (retrospective of Assessment model in MSE)*

Description

Plots the true retrospective of an assessment model during the MSE. A series of time series estimates of SSB, F, and VB are plotted over the course of the MSE are plotted against the operating model (true) values (in black).

Usage

```
retrospective_AM(MSE, sim = 1, MP, MSE_Hist = NULL, plot_legend = FALSE)
```

Arguments

MSE	An object of class MSE created by runMSE with PPD = TRUE.
sim	Integer between 1 and MSE@nsim. The simulation number for which the retrospectives will be plotted.
MP	Character. The name of the management procedure created by make_MP containing the asssessment model.
MSE_Hist	Optional. The list containing historical data for the MSE, created by runMSE with argument Hist = TRUE. Currently only used to plot operating model vulnerable biomass in historical period.
plot_legend	Logical. Whether to plot legend to reference year of assessment in the MSE.

Details

For assessment models that utilize annual harvest rates (u), the instantaneous fishing mortality rates are obtained as F = -log(1 - u).

Value

A series of figures for spawning stock biomass (SSB, including absolute magnitude and relative to MSY and virgin), fishing mortality (F, including absolute magnitude and relative to MSY), and vulnerable biomass (VB) estimates over the course of the MSE are plotted against the operating model (true) values (in black).

Note

This function only plots retrospectives from a single simulation in the MSE. Results from one figure may not be indicative of general assessment behavior and performance overall.

For SP and SP_SS assessment models don't model SSB. Instead, the estimated vulnerable biomass is plotted.

Author(s)

Q. Huynh

See Also

diagnostic_AM

Examples

```
## Not run:
DD_MSY <- makeMP(DD_TMB, HCR_MSY, diagnostic = "full")
myMSE_hist <- DLMtool::runMSE(DLMtool::testOM, Hist = TRUE)
myMSE <- DLMtool::runMSE(DLMtool::testOM, MPs = "DD_MSY", PPD = TRUE)
retrospective_AM(myMSE, sim = 1, MP = "DD_MSY")
retrospective_AM(myMSE, sim = 1, MP = "DD_MSY", Hist = myMSE_hist)

## End(Not run)
```

SCA *Statistical catch-at-age (SCA) model*

Description

A generic statistical catch-at-age model (single fleet, single season) that uses catch, index, and catch-at-age composition data. SCA parameterizes R0 and steepness as leading productivity parameters in the assessment model. Recruitment is estimated as deviations from the resulting stock-recruit relationship. In SCA2, the mean recruitment in the time series is estimated and recruitment deviations around this mean are estimated as penalized parameters (similar to Cadigan 2016). The standard deviation is set high so that the recruitment is almost like free parameters. Unfished and MSY reference points are inferred afterwards from the assessment output (SSB and recruitment estimates). SCA_Pope is a variant of SCA that fixes the expected catch to the observed catch, and Pope's approximation is used to calculate the annual harvest rate (U).

Usage

```
SCA(
  x = 1,
  Data,
  SR = c("BH", "Ricker"),
  vulnerability = c("logistic", "dome"),
  CAA_dist = c("multinomial", "lognormal"),
  CAA_multiplier = 50,
  I_type = c("B", "VB", "SSB"),
  rescale = "mean1",
  max_age = Data@MaxAge,
  start = NULL,
  fix_h = TRUE,
  fix_F_equilibrium = TRUE,
  fix_omega = TRUE,
  fix_sigma = FALSE,
  fix_tau = TRUE,
  early_dev = c("comp_onegen", "comp", "all"),
  late_dev = "comp50",
  integrate = FALSE,
  silent = TRUE,
  opt_hess = FALSE,
  n_restart = ifelse(opt_hess, 0, 1),
  control = list(iter.max = 2e+05, eval.max = 4e+05),
  inner.control = list(),
  ...
)

SCA2(
  x = 1,
  Data,
```

```
  SR = c("BH", "Ricker"),
  vulnerability = c("logistic", "dome"),
  CAA_dist = c("multinomial", "lognormal"),
  CAA_multiplier = 50,
  I_type = c("B", "VB", "SSB"),
  rescale = "mean1",
  max_age = Data@MaxAge,
  start = NULL,
  fix_h = TRUE,
  fix_F_equilibrium = TRUE,
  fix_omega = TRUE,
  fix_sigma = FALSE,
  fix_tau = TRUE,
  common_dev = "comp50",
  integrate = FALSE,
  silent = TRUE,
  opt_hess = FALSE,
  n_restart = ifelse(opt_hess, 0, 1),
  control = list(iter.max = 2e+05, eval.max = 4e+05),
  inner.control = list(),
  ...
)

SCA_Pope(
  x = 1,
  Data,
  SR = c("BH", "Ricker"),
  vulnerability = c("logistic", "dome"),
  CAA_dist = c("multinomial", "lognormal"),
  CAA_multiplier = 50,
  I_type = c("B", "VB", "SSB"),
  rescale = "mean1",
  max_age = Data@MaxAge,
  start = NULL,
  fix_h = TRUE,
  fix_U_equilibrium = TRUE,
  fix_sigma = FALSE,
  fix_tau = TRUE,
  early_dev = c("comp_onegen", "comp", "all"),
  late_dev = "comp50",
  integrate = FALSE,
  silent = TRUE,
  opt_hess = FALSE,
  n_restart = ifelse(opt_hess, 0, 1),
  control = list(iter.max = 2e+05, eval.max = 4e+05),
  inner.control = list(),
  ...
)
```

Arguments

x	A position in the Data object (by default, equal to one for assessments).
Data	An object of class Data
SR	Stock-recruit function (either "BH" for Beverton-Holt or "Ricker").
vulnerability	Whether estimated vulnerability is "logistic" or "dome" (double-normal). See details for parameterization.
CAA_dist	Whether a multinomial or lognormal distribution is used for likelihood of the catch-at-age matrix. See details.
CAA_multiplier	Numeric for data weighting of catch-at-age matrix if CAA_hist = "multinomial". Otherwise ignored. See details.
I_type	Whether the index surveys population biomass (B; this is the default in the DLMtool operating model), vulnerable biomass (VB), or spawning stock biomass (SSB).
rescale	A multiplicative factor that rescales the catch in the assessment model, which can improve convergence. By default, "mean1" scales the catch so that time series mean is 1, otherwise a numeric. Output is re-converted back to original units.
max_age	Integer, the maximum age (plus-group) in the model.
start	Optional list of starting values. Entries can be expressions that are evaluated in the function. See details.
fix_h	Logical, whether to fix steepness to value in Data@steep in the model for SCA. This only affects calculation of reference points for SCA2.
fix_F_equilibrium	Logical, whether the equilibrium fishing mortality prior to the first year of the model is estimated. If TRUE, F_equilibrium is fixed to value provided in start (if provided), otherwise, equal to zero (assumes unfished conditions).
fix_omega	Logical, whether the standard deviation of the catch is fixed. If TRUE, sigma is fixed to value provided in start (if provided), otherwise, value based on Data@CV_Cat.
fix_sigma	Logical, whether the standard deviation of the index is fixed. If TRUE, sigma is fixed to value provided in start (if provided), otherwise, value based on Data@CV_Ind.
fix_tau	Logical, the standard deviation of the recruitment deviations is fixed. If TRUE, tau is fixed to value provided in start (if provided), otherwise, value based on Data@sigmaR.
early_dev	Numeric or character string describing the years for which recruitment deviations are estimated in SCA. By default, equal to "comp_onegen", where rec devs are estimated one full generation prior to the first year when catch-at-age (CAA) data are available. With "comp", rec devs are estimated starting in the first year with CAA. With "all", rec devs start at the beginning of the model. If numeric, the number of years after the first year of the model for which to start estimating rec devs. Use negative numbers for years prior to the first year.

late_dev — Typically, a numeric for the number of most recent years in which recruitment deviations will not be estimated in SCA (recruitment in these years will be based on the mean predicted by stock-recruit relationship). By default, "comp50" uses the number of ages (smaller than the mode) for which the catch-at-age matrix has less than half the abundance than that at the mode.

integrate — Logical, whether the likelihood of the model integrates over the likelihood of the recruitment deviations (thus, treating it as a random effects/state-space variable). Otherwise, recruitment deviations are penalized parameters.

silent — Logical, passed to MakeADFun, whether TMB will print trace information during optimization. Used for dignostics for model convergence.

opt_hess — Logical, whether the hessian function will be passed to nlminb during optimization (this generally reduces the number of iterations to convergence, but is memory and time intensive and does not guarantee an increase in convergence rate). Ignored if integrate = TRUE.

n_restart — The number of restarts (calls to nlminb) in the optimization procedure, so long as the model hasn't converged. The optimization continues from the parameters from the previous (re)start.

control — A named list of agruments for optimization to be passed to nlminb.

inner.control — A named list of arguments for optimization of the random effects, which is passed on to newton.

... — Other arguments to be passed.

common_dev — Typically, a numeric for the number of most recent years in which a common recruitment deviation will be estimated (in SCA2, uninformative years will have a recruitment closer to the mean, which can be very misleading, especially near the end of the time series). By default, "comp50" uses the number of ages (smaller than the mode) for which the catch-at-age matrix has less than half the abundance than that at the mode.

fix_U_equilibrium — Logical, same as 'fix_F_equilibrium' for 'SCA_Pope'.

Details

The basic data inputs are catch (by weight), index (by weight/biomass), and catch-at-age matrix (by numbers). Annual F's are estimated parameters assuming continuous fishing over the year. Note: prior to version 1.2, catches were assumed to be known perfectly with an annual harvest rate from pulse fishing in SCA. That feature has now moved to SCA_Pope.

By default, steepness is fixed in the model to the value in Data@steep.

The annual sample sizes of the catch-at-age matrix is provided to the model (used in the likelihood for catch-at-age assuming a multinomial distribution), and is manipulated via argument CAA_multiplier. This argument is interpreted in two different ways depending on the value provided. If CAA_multiplier > 1, then this value will cap the annual sample sizes to that number. If CAA_multiplier <= 1, then all the annual samples sizes will be re-scaled by that number. By default, sample sizes are capped at 50.

Alternatively, a lognormal distribution with inverse proportion variance can be used for the catch at age (Punt and Kennedy, 1994, as cited by Maunder 2011).

For start (optional), a named list of starting values of estimates can be provided for:

- R0 Virgin recruitment, only for SCA.
- h Steepness, only for SCA. If not provided, the value in Data@steep is used.
- meanR Mean recruitment, only for SCA2.
- F_equilibrium Fishing mortality prior to the first year of model, e.g. zero means unfished conditions. Defaults to zero.
- vul_par Vulnerability parameters (length 2 vector for logistic or length 4 for dome, see below). Users should provide estimates of the parameters in normal space, e.g. vul_max between 0-1, and the function will perform the appropriate transformations for the model.
- F A vector of F's of length nyears, length(Data@Year). If not provided, defaults to 0.1.
- omega Standard deviation of catch. If not provided, the value based on Data@CV_Cat is used.
- sigma Standard deviation of index. If not provided, the value based on Data@CV_Ind is used.
- tau Standard deviation of recruitment deviations. If not provided, the value in Data@sigmaR is used.

Vulnerability can be specified to be either logistic or dome. If logistic, then the parameter vector vul_par is of length 2:

- vul_par[1]: a_95, the age of 95% vulnerability, via logit transformation to constrain a_95 to less than 75% of the maximum age: a_95 = 0.75 * max_age * plogis(vul_par[1]).
- vul_par[2]: a_50, the age of 50% vulnerability as an offset, i.e., a_50 = a_95 -exp(vul_par[2]).

A vague prior for vul_par[2] ~ N(0, sd = 3) is used to aid convergence, for example, when vulnerability » 0.5 for the youngest age class.

With dome vulnerability, a double Gaussian parameterization is used, where vul_par is an estimated vector of length 4:

- vul_par[1]: a_asc, the first age of full vulnerability for the ascending limb, via logit transformation to constrain a_95 to less than 75% of the maximum age: a_asc = 0.75 * maxage * plogis(vul_par[1]).
- vul_par[2]: a_50, the age of 50% vulnerability for the ascending limb as an offset, i.e., a_50 = a_asc -exp(vul_par[2]).
- vul_par[3]: a_des, the last age of full vulnerability (where the descending limb starts) via logit transformation to constrain between a_asc and max_age, i.e., a_des = (max_age -a_asc) * plogis(vul_par[3]) + a_asc. By default, fixed to a small value so that the dome is effectively a three-parameter function.
- vul_par[4]: vul_max, the vulnerability (in logit space) at the maximum age.

Vague priors of vul_par[2] ~ N(0, sd = 3) and vul_par[3] ~ N(0, 3) are used to aid convergence, for example, when vulnerability » 0.5 for the youngest age class.

Value

An object of class Assessment.

Required Data

- SCA, SCA_Pope, and SCA_Pope: Cat, Ind, Mort, L50, L95, CAA, vbK, vbLinf, vbt0, wla, wlb, MaxAge

Optional Data

- SCA: Rec, steep, sigmaR, CV_Ind, CV_Cat
- SC2: Rec, steep, CV_Ind, CV_Cat
- SCA_Pope: Rec, steep, sigmaR, CV_Ind

Author(s)

Q. Huynh

References

Cadigan, N.G. 2016. A state-space stock assessment model for northern cod, including under-reported catches and variable natural mortality rates. Canadian Journal of Fisheries and Aquatic Science 72:296-308.

Maunder, M.N. 2011. Review and evaluation of likelihood functions for composition data in stock-assessment models: Estimating the effective sample size. Fisheries Research 209:311-319.

Punt, A.E. and Kennedy, R.B. 1997. Population modelling of Tasmanian rock lobster, Jasus edwardsii, resources. Marine and Freshwater Research 48:967-980.

See Also

plot.Assessment summary.Assessment retrospective profile make_MP

Examples

```
res <- SCA(Data = DLMtool::SimulatedData)
res2 <- SCA2(Data = DLMtool::SimulatedData)

compare_models(res, res2)

SCA_assess <- SCA2(Data = DLMtool::Simulation_1)

## Not run:
plot(res)

## End(Not run)
```

SIL *Slot in list: get the slot values from a list of objects*

Description

Create of vector of values that correspond with a slot in a list of objects

Usage

```
SIL(listy, sloty)
```

Arguments

listy	A list of objects
sloty	A character vector representing the slot name

Author(s)

T. Carruthers

`simmov` *Calculates movement matrices from user inputs*

Description

A wrapper function for makemov used to generate movement matrices for a DLMtool operating model. Calculates a movement matrix from user-specified unfished stock biomass fraction in each area and probability of staying in the area in each time step.

Usage

```
simmov(
  OM,
  dist = c(0.1, 0.2, 0.3, 0.4),
  prob = 0.5,
  distE = 0.1,
  probE = 0.1,
  prob2 = NA,
  figure = TRUE
)

plot_mov(mov, age = 1, type = c("matrix", "all"))
```

Arguments

OM	Operating model, an object of class OM.
dist	A vector of fractions of unfished stock in each area. The length of this vector will determine the number of areas (nareas) in the OM.
prob	Mean probability of staying across all areas (single value) or a vector of the probability of individuals staying in each area (same length as dist)
distE	Logit (normal) St.Dev error for sampling stock fractions from the fracs vector
probE	Logit (normal) St.Dev error for sampling desired probability of staying either by area (prob is same length as dist) or the mean probability of staying (prob is a single number)
prob2	Optional vector as long as prob and dist. Upper bounds on uniform sampling of probability of staying, lower bound is prob.

figure	Logical to indicate if the movement matrix will be plotted (mean values and range across OM@nsim simulations.)
mov	A four-dimensional array of dimension c(nsim,maxage,nareas,nareas) specifying movement in the operating model.
age	An age from 1 to maxage for the movement-at-age matrix figure when type = "matrix".
type	Whether to plot a movement matrix for a single age ("matrix") or the full movement versus age figure ("all")

Value

The operating model OM with movement parameters in slot cpars. The mov array is of dimension nsim, maxage, nareas, nareas.

Functions

- simmov: Estimation function for creating movement matrix.
- plot_mov: Plotting function.

Note

Array mov is age-specific, but currently the movement generated by simmov is independent of age.

Author(s)

T. Carruthers and Q. Huynh

Examples

```
movOM_5areas <- simmov(testOM, dist = c(0.01,0.1,0.2,0.3,0.39), prob = c(0.1,0.6,0.6,0.7,0.9))
movOM_5areas@cpars$mov[1, 1, , ] # sim 1, age 1, movement from areas in column i to areas in row j
plot_mov(movOM_5areas@cpars$mov)
plot_mov(movOM_5areas@cpars$mov, type = "all")
```

SOL *Sum over list: get the list values from a list of lists*

Description

Create of vector of values that correspond with a named position in a list of objects

Usage

```
SOL(listy, namey)
```

Arguments

listy	A list of objects
namey	A character vector representing the list item's name

Author(s)

T. Carruthers

SP *Surplus production model with FMSY and MSY as leading parameters*

Description

A surplus production model that uses only a time-series of catches and a relative abundance index and coded in TMB. The base model, SP, is conditioned on catch and estimates a predicted index. Continuous surplus production and fishing is modeled with sub-annual time steps which should approximate the behavior of ASPIC (Prager 1994). The Fox model, SP_Fox, fixes BMSY/K = 0.37 (1/e). The state-space version, SP_SS estimates annual deviates in biomass. An option allows for setting a prior for the intrinsic rate of increase. The function for the spict model (Pedersen and Berg, 2016) is available in DLMextra.

Usage

```
SP(
  x = 1,
  Data,
  rescale = "mean1",
  start = NULL,
  fix_dep = TRUE,
  fix_n = TRUE,
  n_seas = 4L,
  n_itF = 3L,
  use_r_prior = FALSE,
  r_reps = 100,
  SR_type = c("BH", "Ricker"),
  silent = TRUE,
  opt_hess = FALSE,
  n_restart = ifelse(opt_hess, 0, 1),
  control = list(iter.max = 5000, eval.max = 10000),
  ...
)

SP_SS(
  x = 1,
  Data,
  rescale = "mean1",
```

```
  start = NULL,
  fix_dep = TRUE,
  fix_n = TRUE,
  fix_sigma = TRUE,
  fix_tau = TRUE,
  early_dev = c("all", "index"),
  n_seas = 4L,
  n_itF = 3L,
  use_r_prior = FALSE,
  r_reps = 100,
  SR_type = c("BH", "Ricker"),
  integrate = FALSE,
  silent = TRUE,
  opt_hess = FALSE,
  n_restart = ifelse(opt_hess, 0, 1),
  control = list(iter.max = 5000, eval.max = 10000),
  inner.control = list(),
  ...
)

SP_Fox(x = 1, Data, ...)
```

Arguments

x	An index for the objects in Data when running in runMSE. Otherwise, equals to 1 When running an assessment interactively.
Data	An object of class Data.
rescale	A multiplicative factor that rescales the catch in the assessment model, which can improve convergence. By default, "mean1" scales the catch so that time series mean is 1, otherwise a numeric. Output is re-converted back to original units.
start	Optional list of starting values. Entries can be expressions that are evaluated in the function. See details.
fix_dep	Logical, whether to fix the initial depletion (ratio of biomass to carrying capacity in the first year of the model). If TRUE, uses the value in start, otherwise equal to 1 (unfished conditions).
fix_n	Logical, whether to fix the exponent of the production function. If TRUE, uses the value in start, otherwise equal to n = 2, where the biomass at MSY is half of carrying capacity.
n_seas	Integer, the number of seasons in the model for calculating continuous surplus production.
n_itF	Integer, the number of iterations to solve F conditional on the observed catch given multiple seasons within an annual time step. Ignored if n_seas = 1.
use_r_prior	Logical, whether a prior for the intrinsic rate of increase will be used in the model. See details.
r_reps	If use_r_prior = TRUE, the number of samples of natural mortality and steepness for calculating the mean and standard deviation of the r prior.

SR_type	If use_r_prior = TRUE, the stock-recruit relationship used to calculate unfished recruits per spawner at the origin of spwaning biomass approaches zero. Used for the r prior.
silent	Logical, passed to MakeADFun, whether TMB will print trace information during optimization. Used for dignostics for model convergence.
opt_hess	Logical, whether the hessian function will be passed to nlminb during optimization (this generally reduces the number of iterations to convergence, but is memory and time intensive and does not guarantee an increase in convergence rate). Ignored if integrate = TRUE.
n_restart	The number of restarts (calls to nlminb) in the optimization procedure, so long as the model hasn't converged. The optimization continues from the parameters from the previous (re)start.
control	A named list of parameters regarding optimization to be passed to nlminb.
...	For SP_Fox, additional arguments to pass to SP.
fix_sigma	Logical, whether the standard deviation of the index is fixed. If TRUE, sigma is fixed to value provided in start (if provided), otherwise, value based on Data@CV_Ind.
fix_tau	Logical, the standard deviation of the biomass deviations is fixed. If TRUE, tau is fixed to value provided in start (if provided), otherwise, equal to 0.1.
early_dev	Character string describing the years for which biomass deviations are estimated in SP_SS. By default, deviations are estimated in each year of the model ("all"), while deviations could also be estimated once index data are available ("index").
integrate	Logical, whether the likelihood of the model integrates over the likelihood of the biomass deviations (thus, treating it as a state-space variable).
inner.control	A named list of arguments for optimization of the random effects, which is passed on to to newton via MakeADFun.

Details

To provide starting values for the SP, a named list can be provided for FMSY, MSY, dep, and n via the start argument (see example).

For SP_SS, a start value can also be provided for sigma and tau, the standard deviation of the index and log-biomass deviates, respectively. Default for tau is 0.1. Deviations are estimated beginning in the year when index data are available.

If use_r_prior = TRUE, SP and SP_SS will use a prior for the intrinsic rate of increase in the objective function. A vector of length two can be passed in the start list for the mean and standard deviation of the prior (see example). The normal distribution is used.

If no values are provided, a prior is created using the Euler-Lotka method (Equation 15a of McAllister et al. 2001). The Euler-Lotka method is modified to multiply the left-hand side of equation 15a by the alpha parameter of the stock-recruit relationship (Stanley et al. 2009). Natural mortality and steepness are sampled in order to generate a prior distribution for r. See vignette("Surplus_production") for more details.

Value

An object of Assessment containing objects and output from TMB.

Required Data

- SP: Cat, Ind
- SP_SS: Cat, Ind

Optional Data

SP_SS: CV_Ind

Note

The model uses the Fletcher (1978) formulation and is parameterized with FMSY and MSY as leading parameters. The default conditions assume unfished conditions in the first year of the time series and a symmetric production function (n = 2).

Tip: to create the Fox model (Fox 1970), just fix n = 1. See example.

Author(s)

Q. Huynh

References

Fletcher, R. I. 1978. On the restructuring of the Pella-Tomlinson system. Fishery Bulletin 76:515:521.

Fox, W.W. 1970. An exponential surplus-yield model for optimizing exploited fish populations. Transactions of the American Fisheries Society 99:80-88.

McAllister, M.K., Pikitch, E.K., and Babcock, E.A. 2001. Using demographic methods to construct Bayesian priors for the intrinsic rate of increase in the Schaefer model and implications for stock rebuilding. Can. J. Fish. Aquat. Sci. 58: 1871-1890.

Pedersen, M. W. and Berg, C. W. 2017. A stochastic surplus production model in continuous time. Fish and Fisheries. 18:226-243.

Pella, J. J. and Tomlinson, P. K. 1969. A generalized stock production model. Inter-Am. Trop. Tuna Comm., Bull. 13:419-496.

Prager, M. H. 1994. A suite of extensions to a nonequilibrium surplus-production model. Fishery Bulletin 92:374-389.

Stanley, R.D., M. McAllister, P. Starr and N. Olsen. 2009. Stock assessment for bocaccio (Sebastes paucispinis) in British Columbia waters. DFO Can. Sci. Advis. Sec. Res. Doc. 2009/055. xiv + 200 p.

See Also

SP_production plot.Assessment summary.Assessment retrospective profile make_MP

Examples

```
data(swordfish)

#### Observation-error surplus production model
res <- SP(Data = swordfish)

# Provide starting values, assume B/K = 0.875 in first year of model
# and symmetrical production curve (n = 2)
start <- list(dep = 0.875, n = 2)
res <- SP(Data = swordfish, start = start)

## Not run:
plot(res)

## End(Not run)

profile(res, FMSY = seq(0.1, 0.4, 0.01))
retrospective(res)


#### State-space version
res_SS <- SP_SS(Data = swordfish, start = list(dep = 0.875, sigma = 0.1, tau = 0.1))

## Not run:
plot(res_SS)

## End(Not run)

#### Fox model
res_Fox <- SP(Data = swordfish, start = list(n = 1), fix_n = TRUE)
res_Fox2 <- SP_Fox(Data = swordfish)

#### SP with r_prior
res_prior <- SP(Data = SimulatedData, use_r_prior = TRUE)

#### Pass an r_prior to the model with mean = 0.35, sd = 0.10
res_prior2 <- SP(Data = SimulatedData, use_r_prior = TRUE, start = list(r_prior = c(0.35, 0.10)))
```

SP_production *Find the production parameter based on depletion that produces MSY*

Description

For surplus production models, this function returns the production exponent n corresponding to BMSY/K (Fletcher 1978).

Usage

```
SP_production(depletion, figure = TRUE)
```

Arguments

depletion	The hypothesized depletion that produces MSY.
figure	Local, plots figure of production function as a function of depletion (B/K)

Value

The production function exponent n (numeric).

Note

May be useful for parameterizing n in SP and SP_SS.

Author(s)

Q. Huynh

References

Fletcher, R. I. 1978. On the restructuring of the Pella-Tomlinson system. Fishery Bulletin 76:515:521.

See Also

SP SP_SS

Examples

```
SP_production(0.5)
SP_production(0.5)
```

SRA-class *Class*-SRA

Description

An S4 class for the output from SRA_scope.

Slots

- OM An updated operating model, class OM.
- SSB A matrix of estimated spawning biomass with OM@nsim rows and OM@nyears+1 columns.
- NAA An array for the predicted numbers at age with dimension OM@nsim, OM@nyears+1, and OM@maxage.
- CAA An array for the predicted catch at age with dimension OM@nsim, OM@nyears, OM@maxage, and nfleet.
- CAL An array for the predicted catch at length with dimension OM@nsim, OM@nyears, length bins, and nfleet.

`conv` A logical vector of length `OM@nsim` indicating convergence of the SRA scoping model in the i-th simulation.

`Misc` A list of length `OM@nsim` with more output from the fitted SRA scoping model.

`mean_fit` A list of output from fit to mean values of life history parameters in the operating model.

`data` A list of the data inputs for the SRA scoping model.

`config` A data frame describing configuration of the SRA scoping model.

Author(s)

Q. Huynh

See Also

plot.SRA SRA_scope

`SRA_scope` *Stock-reduction analysis (SRA) for conditioning operating models*

Description

Intended for conditioning operating models for data-limited stocks. From a historical time series of total catch or effort, and potentially age/length compositions and multiple indices of abundance, the SRA returns a range of values for depletion, selectivity, unfished recruitment (R0), historical fishing effort, and recruitment deviations for the operating model. This is done by sampling life history parameters provided by the user and fitting to the data in a statistical catch-at-age model (with the predicted catch equal to the observed catch). This function is intended to generate a range of potential depletion scenarios that could be supported from sparse data. Either a full catch (conditioned on catch) or effort (conditioned on effort) time series is needed but missing data (as NAs) are allowed for all other data types.

Usage

```
SRA_scope(
  OM,
  data = list(),
  condition = c("catch", "effort"),
  selectivity = "logistic",
  s_selectivity = NULL,
  LWT = list(),
  comp_like = c("multinomial", "lognormal"),
  ESS = c(30, 30),
  max_F = 3,
  cores = 1L,
  integrate = FALSE,
  mean_fit = FALSE,
  drop_nonconv = FALSE,
```

```
  ...
)

plot_SRA_scope(...)

Sub_cpars(OM, sims = 1:OM@nsim)
```

Arguments

OM	An object of class OM that specifies natural mortality (M), growth (Linf, K, t0, a, b), stock-recruitment relationship, steepness, maturity parameters (L50 and L50_95), standard deviation of recruitment variability (Perr), as well as index uncertainty (Iobs).
data	A list of data inputs. See Data section below.
condition	String to indicate whether the SRA model is conditioned on catch or effort.
selectivity	A character vector of length nfleet to indicate "logistic" or "dome" selectivity for each fleet in Chist.
s_selectivity	A vector of length nsurvey to indicate "logistic" or "dome" selectivity for each survey in Index. Use a number for an age-specific index. Only used if any of the corresponding entries of data$I_type = "est" or if a number is specified here.
LWT	A named list of likelihood weights for the SRA model. See details.
comp_like	A string indicating either "multinomial" (default) or "lognormal" distributions for the composition data.
ESS	If comp_like = "multinomial", a numeric vector of length two to cap the maximum effective samples size of the age and length compositions, respectively, for the multinomial likelihood function. The effective sample size of an age or length composition sample is the minimum of ESS or the number of observations (sum across columns). For more flexibility, set ESS to be very large and alter the arrays as needed.
max_F	The maximum F for any fleet in the scoping model (higher F's in the model are penalized in the objective function).
cores	Integer for the number of CPU cores for the stock reduction analysis.
integrate	Logical, whether to treat recruitment deviations as penalized parameters (FALSE) or random effects (TRUE).
mean_fit	Logical, whether to run an additional with mean values of life history parameters from the OM.
drop_nonconv	Logical, whether to drop non-converged fits of the SRA model.
...	Other arguments to pass in for starting values of parameters and fixing parameters. See details.
sims	A logical vector of length OM@nsim or a numberic vector indicating which simulations to keep.

Details

For SRA_scope, additional arguments can be passed to the model via . . . :

- vul_par: A matrix of 3 rows and nfleet columns for starting values for fleet selectivity. The three rows correspond to L5 (length of 5 percent selectivity), LFS (length of full selectivity) and Vmaxlen (selectivity at length Linf). By default, the starting values are values from the OM object.
- s_vul_par: A matrix of 3 rows and nsurvey columns for starting values for fleet selectivity. Same setup as vul_par.
- map_vul_par: The map argument for vul_par in TMB, see MakeADFun, which indicates whether selectivity parameters are fixed or estimated. A matrix of the same dimension as vul_par. If an entry is NA, the corresponding parameter is fixed in the model to the starting value. Otherwise, an integer for each independent parameter. By default, selectivity is fixed if there are no age or length composition for that fleet or survey, otherwise estimated.
- map_s_vul_par: The map argument for the survey selectivity parameters (same dimension as s_vul_par).
- map_log_rec_dev: A vector of length OM@nyears that indexes which recruitment deviates are fixed (using NA) or estimated (a separate integer).

Survey selectivity is estimated only if s_CAA or s_CAL is provided. Otherwise, the selectivity should be mirrored to a fleet (vulnerable biomass selectivity) or indexed to total or spawning biomass (see I_type).

LWT is an optional named list containing the likelihood weights (values > 0) with the possible options:

- Chist: A vector of length nfleet.
- Index: A vector of length nsurvey.
- CAA, CAL, ML, C_eq: A vector of length nfleet for each.
- s_CAA, s_CAL: A vector of length nsurvey for each.

By default, all likelihood weights are equal to one if not specified by the user. Weighting for CAA and CAL can also be adjusted by changing the multinomial sample size. For CAA, CAL, s_CAA, and s_CAL, the arrays should be set up so that the annual number of observations (summed over columns) should be equal to the presumed effective sample size. Argument ESS provides a shortcut to cap the the effective sample size.

plot_SRA_scope is now deprecated in favor of plot.SRA.

Parameters that were used in the fitting model are placed in objects in OM@cpars.

Sub_cpars is a convenient function to subset simulations for the operating model, for example, to remove simulations from unconverged model fits or outlier simulations.

Value

An object of class SRA, including the updated operating model object.

Data

One of indices, age compositions, or length compositions should be provided in addition to the historical catch or effort. Not all arguments are needed to run the model (some have defaults, while others are ignored if not applicable depending on the data provided).

The `data` list can include:

- Chist - A vector of historical catch, should be of length OM@nyears. If there are multiple fleets: a matrix of OM@nyears rows and nfleet columns. Ideally, the first year of the catch series represents unfished conditions (see also `C_eq`).
- Ehist - A vector of historical effort, should be of length OM@nyears (see also `E_eq`).
- Index - A vector of values of an index (of length OM@nyears). If there are multiple surveys: a matrix of historical indices of abundances, with rows indexing years and columns indexing surveys. Age-specific indices should be numbers-specific while all others are weight-based.
- I_sd - A vector or matrix of standard deviations (lognormal distribution) for the indices corresponding to the entries in `Index`. If not provided, this function will use values from `OM@Iobs`.
- I_type - A character vector of length nsurvey to indicate the type of biomass for which each index follows. Either `"B"` for total biomass, or `"SSB"` for spawning biomass. If not provided, "B" is used. Use numbers if the index corresponds to a fleet in `Chist`. Use `"est"` to set survey selectivity to be an independent component of the model, i.e., as an age-specific index or estimated separatel. Note, this generally requires age `s_CAA` or length `s_CAL`compositions.
- CAA - Fishery age composition matrix with nyears rows and OM@maxage columns. If multiple fleets: an array with dimension: nyears, OM@maxage, and nfleets.
- CAL - Fishery Length composition matrix with nyears rows and columns indexing the length bin. If multiple fleets: an array with dimension: nyears, length bins, and nfleets.
- ML - A vector of fishery mean length observations (length OM@nyears), or if multiple fleets: matrix of dimension: nyears and nfleets. Generally, should not be used if `CAL` is also provided, unless mean length and length comps are independently sampled.
- ML_sd - The standard deviation (normal distribution) of the observed mean lengths. If there are multiple fleets, a vector of length nfleet. If not provided, default value is `0.1 * mean(ML)`.
- s_CAA - Survey age composition data, an array of dimension nyears, maxage, nsurvey.
- s_CAL - Survey length composition data, an array of dimension nyears, length(length_bin), nsurvey.
- length_bin - A vector for the midpoints of the length bins for `CAL` and `s_CAL`. All bin widths should be equal in size.
- C_eq - A numeric vector of length nfleet for the equilibrium catch for each fleet in `Chist` prior to the first year of the operating model. Zero (default) implies unfished conditions in year one. Otherwise, this is used to estimate depletion in the first year of the data.
- E_eq - The equilibrium effort for each fleet in `Ehist` prior to the first year of the operating model. Zero (default) implies unfished conditions in year one. Otherwise, this is used to estimate depletion in the first year of the data.
- abs_I - Optional, an integer vector to indicate which indices are in absolute magnitude. Use 1 to set q = 1, otherwise use 0 to estimate q.
- I_basis - Optional, an integer vector to indicate whether indices are biomass based (1) or abundance-based (0). By default, all are biomass-based.

Selectivity is fixed to values sampled from `OM` if no age or length compositions are provided.

Note

If the operating model `OM` uses time-varying growth or M, then those trends will be used in the SRA as well. Time-varying life history parameters create ambiguity in the calculation and interpretation of depletion and reference points in runMSE. See section D.5 of `DLMtool::userguide()`.

The easiest way to turn off time-varying growth/M is by setting: `OM@Msd <-OM@Linfsd <-OM@Ksd <-c(0,0)`.

Author(s)

Q. Huynh

See Also

plot.SRA SRA

`SS2Data` *Reads data Stock Synthesis file structure into a Data object using package r4ss*

Description

A function that uses the file location of a fitted SS3 model including input files to population the various slots of an Data object.

Usage

```
SS2Data(
  SSdir,
  Name = "Object generated by SS2Data function",
  Common_Name = "",
  Species = "",
  Region = "",
  min_age_M = 1,
  gender = 1,
  comp_fleet = "all",
  comp_season = "sum",
  comp_partition = "all",
  comp_gender = "all",
  index_fleet = "SSB",
  index_season = "mean",
  ...
)
```

Arguments

SSdir	A folder with Stock Synthesis input and output files in it
Name	The name for the Data object
Common_Name	Character string for the common name of the stock.
Species	Scientific name of the species
Region	Geographic region of the stock or fishery.
min_age_M	Currently, the Data object supports a single value of M for all ages. The argument selects the minimum age for calculating the mean of age-dependent M from the SS assessment.
gender	An integer index for the sex for which life history values are importing (1 = female, 2 = male).
comp_fleet	A vector of indices corresponding to fleets in the assessment over which to aggregate the composition (catch-at-length and catch-at-age) data. By default, characer string "all" will aggregate across all fleets.
comp_season	Integer, for seasonal models, the season for which the value of the index will be used. By default, "mean" will take the average across seasons.
comp_partition	Integer vector for selecting length/age observations that are retained (2), discarded (1), or both (0). By default, "all" sums over all available partitions.
comp_gender	Integer vector for selecting length/age observations that are female (1), male (2), or both (0), or both scaled to sum to one (3). By default, "all" sums over all gender codes.
index_fleet	Integer for selecting the fleet of the index to put in the Data object. By default, "SSB" will use the relative trend in spawning stock biomass as estimated in the model as the index.
index_season	Integer, for seasonal models, the season for which the value of the index will be used. By default, "mean" will take the average across seasons.
...	Arguments to pass to SS_output

Value

An object of class Data.

Note

Currently supports the version of r4ss on CRAN (v.1.24) and Github (v.1.34-35). Function may be incompatible with other versions of r4ss.

Author(s)

T. Carruthers and Q. Huynh

See Also

SS2OM

SS2OM *Reads MLE estimates from Stock Synthesis file structure into an operating model using package r4ss.*

Description

A function that uses the file location of a fitted SS3 model including input files to population the various slots of an operating model with MLE parameter estimates. The function mainly populates the Stock and Fleet portions of the operating model; the user still needs to parameterize most of the observation and implementation portions of the operating model.

Usage

```
SS2OM(
  SSdir,
  nsim = 48,
  proyears = 50,
  reps = 1,
  maxF = 3,
  seed = 1,
  interval = 1,
  Obs = DLMtool::Generic_Obs,
  Imp = DLMtool::Perfect_Imp,
  import_mov = TRUE,
  gender = 1:2,
  age_rec = 1,
  silent = FALSE,
  Name = "OM generated by SS2OM function",
  Source = "No source provided",
  Author = "No author provided",
  report = FALSE,
  filename = "SS2OM",
  dir = tempdir(),
  open_file = TRUE,
  ...
)
```

Arguments

SSdir	A folder with Stock Synthesis input and output files in it.
nsim	The number of simulations to take for parameters with uncertainty (for OM@cpars custom parameters).
proyears	The number of projection years for MSE
reps	The number of stochastic replicates within each simulation in the operating model.
maxF	The maximum allowable F in the operating model.

seed	The random seed for the operating model.
interval	The interval at which management procedures will update the management advice in runMSE, e.g., 1 = annual updates.
Obs	The observation model (class Obs). This function only updates the catch and index observation error.
Imp	The implementation model (class Imp). This function does not update implementation parameters.
import_mov	Logical, whether to import movement matrix from the assessment.
gender	An integer that indexes the sex for importing life history parameters (1 = usually female, 2 = usually male, 1:2 = mean across both sexes).
age_rec	Integer for the age of recruitment. The default is 1 for DLMtool operating models. Generally, should not be changed.
silent	Whether to silence messages to the console.
Name	The name of the operating model
Source	Reference to assessment documentation e.g. a url
Author	Who did the assessment
report	Logical, if TRUE, the function will run runMSE to generate historical data from the operating model to compare against SS output. A markdown report will be generated.
filename	If report = TRUE, character string for the name of the markdown and HTML files.
dir	If report = TRUE, the directory in which the markdown and HTML files will be saved.
open_file	If report = TRUE, whether the HTML document is opened after it is rendered.
...	Arguments to pass to SS_output.

Details

The function generally uses values from the terminal year of the assessment for most life history parameters (maturity, M, etc). This function does detect time-varying growth in the assessment and uses annual length/weight-at-age for historical years. Selectivity is derived from the F-at-age matrix.

Value

An object of class OM.

Note

Currently supports versions of r4ss on CRAN (v.1.24) and Github (v.1.34-35). Function may be incompatible with other versions of r4ss.

Author(s)

T. Carruthers and Q. Huynh

See Also

SS2Data

`summary.Assessment` *Summary of Assessment object*

Description

Returns a summary of parameter estimates and output from an Assessment object.

Usage

```
## S4 method for signature 'Assessment'
summary(object)
```

Arguments

`object`	An object of class Assessment

Value

A list of parameters.

Examples

```
output <- DD_TMB(Data = DLMtool::Simulation_1)
summary(output)
```

`swordfish` *North Atlantic Swordfish dataset*

Description

An S4 object containing catch and index time series for North Atlantic swordfish.

Usage

```
swordfish
```

Format

An object of class Data.

Source

ASPIC Software at https://www.mhprager.com/aspic.html

Examples

```
data(swordfish)
```

TAC_MSY *Calculate MSY-based TAC from Assessment object*

Description

A function to calculate the total allowable catch (TAC). Based on the MSY (maximum sustainable yield) principle, the TAC is the product of either UMSY or FMSY and the available biomass, i.e. vulnerable biomass, in terminal year.

Usage

```
TAC_MSY(Assessment, reps, MSY_frac = 1)

calculate_TAC(Assessment, reps, MSY_frac = 1)
```

Arguments

Assessment	An Assessment object with estimates of UMSY or FMSY and terminal year vulnerable biomass.
reps	The number of stochastic draws of UMSY or FMSY.
MSY_frac	The fraction of FMSY or UMSY for calculating the TAC (e.g. MSY_frac = 0.75 fishes at 75% of FMSY).

Value

A vector of length reps of stochastic samples of TAC recommendation. Returns NA's if missing either UMSY/FMSY or vulnerable biomass.

Note

calculate_TAC is deprecated as of version 1.2 in favor of TAC_MSY because the latter has a more informative name.

See Also

HCR_MSY HCR40_10 HCR60_20

TEG *Toms expand grid*

Description

Create an indexing grid from just a vector of maximum dimension sizes

Usage

```
TEG(vec)
```

Arguments

vec	A vector of maximum array sizes

Author(s)

T. Carruthers

userguide *Get the MSEtool vignettes*

Description

A convenient function to open a web browser with the MSEtool package vignettes

Usage

```
userguide()
```

See Also

userguide

Examples

```
## Not run:
MSEtool::userguide()
DLMtool::userguide()

## End(Not run)
```

VPA *Virtual population analysis (VPA)*

Description

A VPA model that back-calculates abundance-at-age assuming that the catch-at-age is known without error and tuned to an index. The population dynamics equations are primarily drawn from VPA-2BOX (Porch 2018). MSY reference points are then calculated from the VPA output.

Usage

```
VPA(
  x = 1,
  Data,
  expanded = FALSE,
  SR = c("BH", "Ricker"),
  vulnerability = c("logistic", "dome", "free"),
  I_type = c("B", "VB", "SSB"),
  rescale = "mean1",
  start = NULL,
  fix_h = TRUE,
  fix_sigma = FALSE,
  fix_Fratio = TRUE,
  vul_pen = c(3, 0.4),
  R_pen = c(3, Data@sigmaR[x]),
  nitF = 5L,
  silent = TRUE,
  opt_hess = FALSE,
  n_restart = ifelse(opt_hess, 0, 1),
  control = list(iter.max = 2e+05, eval.max = 4e+05),
  ...
)
```

Arguments

x	A position in the Data object (by default, equal to one for assessments).
Data	An object of class Data
expanded	Whether the catch at age in Data has been expanded. If FALSE, then the catch in weight should be provided in Data@Cat so that the function can calculate annual expansion factors.
SR	Stock-recruit function (either "BH" for Beverton-Holt or "Ricker") for calculating MSY reference points.
vulnerability	Whether the terminal year vulnerability is "logistic" or "dome" (double-normal). If "free", independent F's are calculated in the terminal year (subject to the assumed ratio of F of the plus-group to the previous age class). See details for parameterization.

I_type	Whether the index surveys population biomass (B; this is the default in the DLMtool operating model), vulnerable biomass (VB), or spawning stock biomass (SSB).
rescale	A multiplicative factor that rescales the catch in the assessment model, which can improve convergence. By default, "mean1" scales the catch so that time series mean is 1, otherwise a numeric. Output is re-converted back to original units.
start	Optional list of starting values. Entries can be expressions that are evaluated in the function. See details.
fix_h	Logical, whether to fix steepness to value in Data@steep. This only affects calculation of reference points.
fix_sigma	Logical, whether the standard deviation of the index is fixed. If TRUE, sigma is fixed to value provided in start (if provided), otherwise, value based on Data@CV_Ind.
fix_Fratio	Logical, whether the ratio of F of the plus-group to the previous age class is fixed in the model.
vul_pen	A length two vector that parameterizes how the model constrains the vulnerability in the most recent years. The first number is the number of years in which vulnerability will be constrained (as a random walk), the second number is the standard deviation of the random walk.
R_pen	A length two vector that parameterizes how the model constrains the recruitment in the most recent years. The first number is the number of years in which recruitment will be constrained (as a random walk), the second number is the standard deviation of the random walk.
nitF	The number of iterations for solving F in the model (via Newton's method).
silent	Logical, passed to MakeADFun, whether TMB will print trace information during optimization. Used for dignostics for model convergence.
opt_hess	Logical, whether the hessian function will be passed to nlminb during optimization (this generally reduces the number of iterations to convergence, but is memory and time intensive and does not guarantee an increase in convergence rate). Ignored if integrate = TRUE.
n_restart	The number of restarts (calls to nlminb) in the optimization procedure, so long as the model hasn't converged. The optimization continues from the parameters from the previous (re)start.
control	A named list of agruments for optimization to be passed to nlminb.
...	Other arguments to be passed.

Details

The VPA is initialized by estimating the terminal F-at-age. Parameter F_term is the apical terminal F if a functional form for vulnerability is used in the terminal year. If the terminal F-at-age are otherwise independent parameters, F_term is the F for the reference age which is half the maximum age. Once terminal-year abundance is estimated, the abundance in historical years can be back-calculated. The oldest age group is a plus-group, and requires an assumption regarding the ratio of

F's between the plus-group and the next youngest age class. The F-ratio can be fixed (default) or estimated.

For `start` (optional), a named list of starting values of estimates can be provided for:

- `F_term` The terminal year fishing mortality.
- `F_ratio` The ratio of F in the plus-group to the next youngest age. If not provided, a value of 1 is used.
- `vul_par` Vulnerability parameters in the terminal year. This will be of length 2 vector for `"logistic"` or length 4 for `"dome"`, see SCA for further documentation on parameterization. For option `"free"`, this will be a vector of length A-2 where A is the number of age classes in the model. To estimate parameters, vulnerability is initially set to one at half the max age (and subsequently re-calculated relative to the maximum F experienced in that year). Vulnerability in the plus-group is also constrained by the Fratio.
- `sigma` Standard deviation of the index. If not provided, the value based on `Data@CV_Ind` is used.

Value

An object of class Assessment. The F vector is the apical fishing mortality experienced by any age class in a given year. The U vector is the ratio of catch (weight) and vulnerable biomass, which may be a better description of fishing pressure (and UMSY = MSY/VBMSY).

References

Porch, C.E. 2018. VPA-2BOX 4.01 User Guide. NOAA Tech. Memo. NMFS-SEFSC-726. 67 pp.

Index