Package 'SAMtool'

March 9, 2021

Type Package

Title Stock Assessment Methods Toolkit

Version 1.1.2

Date 2021-03-09

Maintainer Quang Huynh <quang@bluematterscience.com>

Description Simulation tools for closed-loop simulation are provided for the 'MSEtool' operating model to inform data-rich fisheries. 'SAMtool' provides a conditioning model, assessment models of varying complexity with standardized reporting, model-based management procedures, and diagnostic tools for evaluating assessments inside closed-loop simulation.

License GPL-3

Depends R (>= 3.5.0), MSEtool (>= 3.0.0)

Imports TMB, corpcor, dplyr, gplots, graphics, methods, rmarkdown, snowfall, stats, utils

LinkingTo TMB, RcppEigen

Suggests caret, covr, ggplot2, knitr, mvtnorm, numDeriv, reshape2, shiny, testthat

LazyData yes

LazyLoad yes

RoxygenNote 7.1.1

Encoding UTF-8

VignetteBuilder knitr

URL https://github.com/Blue-Matter/SAMtool

BugReports https://github.com/Blue-Matter/SAMtool/issues

NeedsCompilation yes

Author Quang Huynh [aut, cre], Tom Carruthers [aut], Adrian Hordyk [aut]

Repository CRAN

Date/Publication 2021-03-09 20:30:08 UTC

R topics documented:

`SAMtool-package` *Stock Assessment Methods Toolkit*

Description

Simulation tools for closed-loop simulation are provided for the 'MSEtool' operating model to inform data-rich fisheries. SAMtool provides an OM conditioning model, assessment models of varying complexity with standardized reporting, diagnostic tools for evaluating assessments within closed-loop simulation, and helper functions for building more complex operating models and model-based management procedures.

How to use SAMtool

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

The following assessment models are available:

- Surplus production (SP, SP_SS, SP_Fox, and `spict`)
- Delay difference (DD_TMB, cDD, DD_SS, and cDD_SS)
- Statistical catch-at-age (SCA, SCA2, and SCA_Pope)
- Simple Stock Synthesis (SSS which implements SCA_Pope with fixed depletion assumption)
- Virtual population analysis (VPA)

The `RCM` (Rapid Conditioning Model) can be used to condition operating models from real data. Information can be found here.

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

Author(s)

Quang Huynh <quang@bluematterscience.com>

Tom Carruthers <tom@bluematterscience.com>

Adrian Hordyk <adrian@bluematterscience.com>

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.

`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` Standard error of U/UMSY.

`SE_F_FMSY` Standard error of F/FMSY.

`SE_B_BMSY` Standard error of B/BMSY.

`SE_B_B0` Standard error of B/B0.

`SE_SSB_SSBMSY` Standard error of SSB/SSBMSY.

`SE_SSB_SSB0` Standard error of SSB/SSB0.

`SE_VB_VBMSY` Standard error of VB/VBMSY.

`SE_VB_VB0` Standard error of VB/VB0.

`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.

`forecast` A list containing components for forecasting:

- `per_recruit` A data frame of SPR (spawning potential ratio) and YPR (yield-per-recruit), calculated for a range of harvest rate of 0 - 0.99 or instantaneous F from 0 - 2.5 FMSY.
- `catch_eq` A function that calculates the catch for the next year (after the model terminal year) when an apical F is provided.

`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 make_MP

Examples

```
output <- DD_TMB(Data = MSEtool::SimulatedData)
class(output)
```

---

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,
  AddInd = "B",
  SR = c("BH", "Ricker"),
  rescale = "mean1",
  start = NULL,
  fix_h = TRUE,
  dep = 1,
  LWT = NULL,
  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,
  AddInd = "B",
  SR = c("BH", "Ricker"),
  rescale = "mean1",
  start = NULL,
  fix_h = TRUE,
  fix_sigma = FALSE,
  fix_tau = TRUE,
  dep = 1,
  LWT = NULL,
  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.
AddInd	A vector of integers or character strings indicating the indices to be used in the model. Integers assign the index to the corresponding index in Data@AddInd, "B" (or 0) represents total biomass in Data@Ind, "VB" represents vulnerable biomass in Data@VInd, and "SSB" represents spawning stock biomass in Data@SpInd.
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.

dep	The initial depletion in the first year of the model. A tight prior is placed on the model objective function to estimate the equilibrium fishing mortality corresponding to the initial depletion. Due to this tight prior, this F should not be considered to be an independent model parameter.
LWT	A vector of likelihood weights for each survey.
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 diagnostics 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

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

- R0 Unfished recruitment. Otherwise, Data@OM$R0[x] is used in closed-loop, and 400% of mean catch otherwise.
- h Steepness. Otherwise, Data@steep[x] is used, or 0.9 if empty.
- F_equilibrium Equilibrium fishing mortality leading into first year of the model (to determine initial depletion). By default, 0.
- tau Lognormal SD of the recruitment deviations (process error) for DD_SS. By default, Data@sigmaR[x].
- sigma Lognormal SD of the index (observation error). By default, Data@CV_Ind[x]. Not used if multiple indices are used.

Multiple indices are supported in the model. Data@Ind, Data@VInd, and Data@SpInd are all assumed to be biomass-based. For Data@AddInd, Data@I_units are used to identify a biomass vs. abundance-based index.

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 = MSEtool::Red_snapper)

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

summary(res@SD) # Parameter estimates
```

---

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(x = 3, Data = MSEtool::SimulatedData)
res2 <- SCA(x = 3, Data = MSEtool::SimulatedData)
res3 <- SP(x = 3, Data = MSEtool::SimulatedData)

compare_models(res, res2, res3)
```

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 can be conditioned on either (1) effort and estimates predicted catch or (2) catch and estimates a predicted index. In the state-space version DD_SS, recruitment deviations from the stock-recruit relationship are estimated.

Usage

```
DD_TMB(
  x = 1,
  Data,
  condition = c("catch", "effort"),
  AddInd = "B",
  SR = c("BH", "Ricker"),
  rescale = "mean1",
  start = NULL,
  fix_h = TRUE,
  dep = 1,
  LWT = NULL,
  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,
  condition = c("catch", "effort"),
  AddInd = "B",
  SR = c("BH", "Ricker"),
  rescale = "mean1",
  start = NULL,
  fix_h = TRUE,
  fix_sd = FALSE,
  fix_tau = TRUE,
  dep = 1,
  LWT = NULL,
  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.
condition	A string to indicate whether to condition the model on catch or effort (ratio of catch and index).
AddInd	A vector of integers or character strings indicating the indices to be used in the model. Integers assign the index to the corresponding index in Data@AddInd, "B" (or 0) represents total biomass in Data@Ind, "VB" represents vulnerable biomass in Data@VInd, and "SSB" represents spawning stock biomass in Data@SpInd.
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.

dep	The initial depletion in the first year of the model. A tight prior is placed on the model objective function to estimate the equilibrium exploitation rate that corresponds to the initial depletion. Due to this tight prior, this F should not be considered to be an independent model parameter.
LWT	A vector of likelihood weights for each survey.
silent	Logical, passed to MakeADFun, whether TMB will print trace information during optimization. Used for diagnostics 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_sd	Logical, whether the standard deviation of the data in the likelihood (index for conditioning on catch or catch for conditioning on effort). If TRUE, the SD is fixed to value provided in start (if provided), otherwise, value based on either Data@CV_Cat or 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 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

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

- R0 Unfished recruitment. Otherwise, Data@OM$R0[x] is used in closed-loop, and 400% of mean catch otherwise.
- h Steepness. Otherwise, Data@steep[x] is used, or 0.9 if empty.
- q_effort Scalar coefficient when conditioning on effort (to scale to F). Otherwise, 1 is the default.
- U_equilibrium Equilibrium harvest rate leading into first year of the model (to determine initial depletion). By default, 0.
- omega Lognormal SD of the catch (observation error) when conditioning on effort. By default, Data@CV_Cat[x].
- tau Lognormal SD of the recruitment deviations (process error) for DD_SS. By default, Data@sigmaR[x].
- sigma Lognormal SD of the index (observation error) when conditioning on catch. By default, Data@CV_Ind[x]. Not used if multiple indices are used.

Multiple indices are supported in the model. Data@Ind, Data@VInd, and Data@SpInd are all assumed to be biomass-based. For Data@AddInd, Data@I_units are used to identify a biomass vs. abundance-based index.

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.

Value

An object of `Assessment` containing objects and output from TMB.

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

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(x = 3, Data = MSEtool::SimulatedData)

# Provide starting values
start <- list(h = 0.95)
res <- DD_TMB(x = 3, Data = MSEtool::SimulatedData, start = start)

summary(res@SD) # Parameter estimates

### State-space version
```

```
### Set recruitment variability SD = 0.3 (since fix_tau = TRUE)
res <- DD_SS(x = 3, Data = MSEtool::SimulatedData, start = list(tau = 0.3))
```

`diagnostic` *Diagnostic of assessments in MSE: did Assess models converge during MSE?*

Description

Diagnostic check for convergence of Assess models during closed-loop simulation. Use when the MP was created with make_MP with argument `diagnostic = "min"` or `"full"`. This function summarizes and plots the diagnostic information.

Usage

```
diagnostic(MSE, MP, gradient_threshold = 0.1, figure = TRUE)

diagnostic_AM(...)
```

Arguments

`MSE`	An object of class MSE created by `runMSE`.
`MP`	Optional, a character vector of MPs that use assessment models.
`gradient_threshold`	The maximum magnitude (absolute value) desired for the gradient of the likelihood.
`figure`	Logical, whether a figure will be drawn.
`...`	Arguments to pass to `diagnostic`.

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

```
OM <- MSEtool::testOM; OM@proyears <- 20
myMSE <- runMSE(OM, MPs = "SCA_4010")
diagnostic(myMSE)
```

`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

HCRlin *Generic linear harvest control rule based on biomass*

Description

A general function used by HCR_ramp that adjusts the output (e.g., F) by a linear ramp based on the value of the OCP relative to target and limit values.

Usage

```
HCRlin(OCP_val, LOCP, TOCP, relF_min = 0, relF_max = 1)
```

Arguments

OCP_val	The value of the operational control point (OCP).
LOCP	Numeric, the limit value for the OCP in the HCR.
TOCP	Numeric, the target value for the OCP in the HCR.
relF_min	The relative maximum value (e.g. a multiple of FMSY) if OCP < LOCP.
relF_max	The relative maximum value (e.g. a multiple of FMSY) if OCP > TOCP.

Value

Numeric 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)
SPMSY <- make_MP(SP, HCR_MSY)

# The MP which fishes at 75% of FMSY
SP75MSY <- make_MP(SP, HCR_MSY, MSY_frac = 0.75)
```

```
myOM <- MSEtool::runMSE(MSEtool::testOM, MPs = c("FMSYref", "SPMSY", "SP75MSY"))
```

HCR_ramp *Linearly ramped harvest control rules*

Description

An output control rule with a ramp that reduces the target F (used for the TAC recommendation) linearly as a function of an operational control point (OCP) such as spawning depletion or spawning biomass relative to that at MSY. The reduction in F is linear when the OCP is between the target OCP (TOCP) and the limit OCP (LOCP). Above the TOCP, the target F is maximized. Below the LOCP, the target F is minimized. For example, the TOCP and LOCP for 40% and 10% spawning depletion, respectively, in the 40-10 control rule. Ftarget is FMSY above the TOCP and zero below the LOCP. Class HCR objects are typically used with function make_MP.

Usage

```
HCR_ramp(
  Assessment,
  reps = 1,
  OCP_type = c("SSB_SSB0", "SSB_SSBMSY"),
  Ftarget_type = c("FMSY", "F01", "Fmax", "FSPR"),
  LOCP = 0.1,
  TOCP = 0.4,
  relF_min = 0,
  relF_max = 1,
  SPR,
  ...
)

HCR40_10(Assessment, reps = 1, Ftarget_type = "FMSY", SPR = 0.4, ...)

HCR60_20(Assessment, reps = 1, Ftarget_type = "FMSY", SPR = 0.4, ...)

HCR80_40MSY(Assessment, reps = 1, Ftarget_type = "FMSY", SPR = 0.4, ...)
```

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.
OCP_type	The type of operational control points (OCPs) for the harvest control rule used to determine the reduction in F. By default, use ("SSB_SSB0" for spawning depletion. Otherwise use "SSB_SSBMSY" for spawning biomass relative to MSY).
Ftarget_type	The type of F used for the target fishing mortality rate.

LOCP	Numeric, the limit value for the OCP in the HCR.
TOCP	Numeric, the target value for the OCP in the HCR.
relF_min	The relative value of Ftarget (i.e., as a proportion) if OCP < LOCP.
relF_max	The relative value of Ftarget if OCP > TOCP.
SPR	The target value of spawning potential ratio if Ftarget_type = "FSPR". By default, 0.4 (F40%).
...	Miscellaneous arguments.

Details

HCR_ramp is the generic ramped-HCR function where user specifies OCP and corresponding limit and target points, as well as minimum and maximum relative F target.

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

HCR80_40MSY uses 0.8 and 0.4 SSBMSY as the LOCP and TOCP, 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 = expression("Operational control point: Estimated"~SSB/SSB[0]),
    ylab = expression(F[target]~~": proportion of"~~F[MSY]),
```

```
    main = "40-10 harvest control rule", type = "l")
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, OCP_type = "SSB_SSB0", LOCP = 0.1, TOCP = 0.4)

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

# 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, OCP_type = "SSB_SSBMSY", LOCP = 0.4, TOCP = 0.8,
relF_min = 0.1, relF_max = 0.75)

# Figure of this conservative HCR
Brel <- seq(0, 1, length.out = 200)
Frel <- HCRlin(Brel, 0.4, 0.8, 0.1, 0.75)
plot(Brel, Frel,
    xlab = expression("Operational control point: Estimated"~SSB/SSB[MSY]),
    ylab = expression(F[target]~":"~~F/F[MSY]),
    ylim = c(0, 1), type = "l")
abline(v = c(0.4, 0.8), col = "red", lty = 2)
```

`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

Value

Density plots of Mahalanobis distance.

Author(s)

T. Carruthers

References

Carruthers and Hordyk 2018

`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("min", "full", "none"), ...)
```

Arguments

`.Assess`	Assessment model, a function of class `Assess`.
`.HCR`	Harvest control rule, a function of class `HCR`.
`diagnostic`	A character string describing if any additional diagnostic information from the assessment models will be collected during the closed-loop simulation (`"none"` is the default). `"min"` (minimal) will collect information on convergence and `"full"` will also collect the model estimates of biomass and F generated by `.Assess`.
`...`	Additional arguments to be passed to `.Assess` and `.HCR`.

Value

A function of class `MP`.

See Also

HCR_ramp HCR_MSY diagnostic retrospective_AM

Examples

```
# A statistical catch-at-age model with a 40-10 control rule
SCA_40_10 <- make_MP(SCA, HCR40_10)

# An SCA that will produce convergence diagnostics
SCA_40_10 <- make_MP(SCA, HCR40_10, diagnostic = "min")

# MP with an SCA that uses a Ricker stock-recruit function.
SCA_Ricker <- make_MP(SCA, HCR_MSY, SR = "Ricker")
show(SCA_Ricker)
```

`Model-based-MP` *Model-based management procedures*

Description

A suite of model-based 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)

SSS_MSY(x, Data, reps = 1)

SSS_75MSY(x, Data, reps = 1)

SSS_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 FMSY, and default arguments for configuring SCA.
- SCA_75MSY: An SCA with a TAC recommendation based on fishing at 75% of FMSY.
- 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 FMSY, 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 FMSY.
- 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 FMSY, and default arguments for configuring SP.
- SP_75MSY: A surplus production model with a TAC recommendation based on fishing at 75% of FMSY.
- SP_4010: A surplus production model with a 40-10 control rule.
- SSS_MSY: Simple stock synthesis (terminal depletion fixed to 0.4 in SSS) with a TAC recommendation based on fishing at FMSY.
- SSS_75MSY: Simple stock synthesis (terminal depletion fixed to 0.4) with with a TAC recommendation based on fishing at 75% FMSY.
- SSS_4010: Simple stock synthesis (terminal depletion fixed to 0.4) with a 40-10 control rule.

Examples

```
MSEtool::avail("MP", package = "SAMtool")


myMSE <- MSEtool::runMSE(MSEtool::testOM, MPs = c("FMSYref", "SCA_4010"))
```

`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,
  render_args = list(),
  ...
)

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

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.
`render_args`	Arguments to pass to render.

...	Other arguments.
y	An object of class retro.

Value

Returns invisibly the output from render.

See Also

retrospective

Examples

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


plot(output)
```

`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.

Value

A likelihood profile plot, either a one-dimensional line plot or a two-dimensional contour plot.

Author(s)

Q. Huynh

plot.RCModel *Plot RCM 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 RCM scoping function, as well as diagnostic plots for the fits to the RCM for each simulation. `compare_RCM` plots a short report that compares output from multiple RCM objects, assuming the same model structure, i.e., identical matrix and array dimensions among models, but different data weightings, data omissions, etc.

Usage

```
## S4 method for signature 'RCModel,missing'
plot(
  x,
  compare = TRUE,
  filename = "RCM",
  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_RCM(
  ...,
  compare = TRUE,
  filename = "compare_RCM",
  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 RCModel (output from RCM).
compare	Logical, if TRUE, the function will run runMSE to compare the historical period of the operating model and the RCM 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 RCM 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_RCM, multiple RCM objects for comparison.

Value

Returns invisibly the output from render.

See Also

RCModel RCM

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.

Value

A series of plots showing retrospective patterns in fishing mortality, spawning biomass, recruitment, etc.

Author(s)

Q. Huynh

Examples

```
res <- SP(Data = swordfish)
ret <- retrospective(res, figure = FALSE)

summary(ret)
plot(ret)
```

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 logit (TRUE) or normal (FALSE) 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,
  bubble_color = c("black", "white"),
  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).
bubble_color	Colors for negative and positive residuals, respectively, for bubble plots.
fit_linewidth	Argument lwd for fitted line.
fit_color	Color of fitted line.

Value

Plots depending on plot_type.

Author(s)

Q. Huynh

Examples

```
plot_composition(obs = SimulatedData@CAA[1, 1:16, ])
plot_composition(obs = SimulatedData@CAA[1, , ], plot_type = "bubble_data",
                 ages = 0:SimulatedData@MaxAge)
```

---

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 lognormal (TRUE) or normal (FALSE) 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.

Value

A plot of model residuals by year (optionally, with error bars).

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 recruitment values 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 <- 0.8
stddev <- 0.1
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.

Value

A plot of annual observed data and predicted values from a model.

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

```
prelim_AM(MSEtool::SimulatedData, SP)
```

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, ...)
```

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

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`
- SSS: `R0`

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 = MSEtool::SimulatedData)

# Profile R0 only
pro <- profile(output, R0 = seq(100, 300, 10))

# Profile both R0 and steepness
pro <- profile(output, R0 = seq(100, 300, 10), h = seq(0.7, 0.9, 0.01))

# 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 simulated observed catch over `p_sim` rows and `p_years` columns.

`Index` An array of simulated observed index of dimension `c(p_sim,p_years,nsurvey)`.

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

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"),
  Ftarget,
  Catch,
  p_years = 50,
  p_sim = 200,
  obs_error,
  process_error,
  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.
Ftarget	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	A list of length two. In the first entry, a vector of length nsurvey giving the standard deviations of each future index, or alternatively an array of dimension p_sim, p_years, and nsurvey giving the deviates. The second entry is the standard deviation of the projected catch. Alternatively, a matrix of simulation and year-specific error structure for the catch (p_sim rows and p_year columns; a matrix of ones indicates perfect data).
process_error	Numeric, standard deviation for process error (e.g., recruitment or biomass deviates). If NULL, uses values from assessment model. Alternatively, a matrix of simulation and year-specific recruitment deviates (p_sim rows and p_year columns, a matrix of ones indicates no recruitment deviates).
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.

Value

An object of class project that contains future predicted values of F, catch, biomass, recruitment, etc.

Examples

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

RCModel-class *Class*-RCModel

Description

An S4 class for the output from RCM.

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+1`.

`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 RCM in the i-th simulation.

`Misc` A list of length `OM@nsim` with more output from the fitted RCM. Within each simulation, items of interest include:

- B - total biomass - vector of length nyears+1
- E0 - annual unfished spawning biomass - vector of length nyears
- E0_SR - unfished spawning biomass for the stock-recruit relationship - numeric
- CR - annual compensation ratio - vector of length nyears
- Arec - alpha parameter of the stock-recruit relationship - numeric
- Brec - beta parameter of the stock-recruit relationship - numeric
- R - recruitment - vector of length nyears+1
- R_early - recruitment for the cohorts in first year of the model - vector maxage-1
- VB - vulnerable biomass - matrix of nyears x nfleet
- N - abundance at age - matrix of nyears+1 x maxage
- F - apical fishing mortality - matrix of nyears x nfleet
- F_at_age - fishing mortality at age - array of nyears x maxage x nfleet
- F_equilibrium - equilibrium fishing mortality prior to first year - vector of length nfleet
- M - natural mortality - matrix of nyears x maxage
- Z - total mortality - matrix of nyears x maxage
- q - survey catchability - vector of length nsurvey
- s_vul - survey selectivity at age - array of dim nyears+1, maxage, nsurvey
- s_vul_len - corresponding survey selectivity at length - matrix of nbins x nsurvey
- Ipred - predicted index values - matrix of nyears x nsurvey
- s_CAApred - predicted survey catch at age - array of dim nyears, maxage, nsurvey
- vul - fleet selectivity at age - array of dim nyears+1, maxage, nfleet (or nsel_block)
- vul_len - corresponding fleet selectivity at length - matrix of nbins x nfleet (or nsel_block)
- s_CALpred - predicted survey catch at length - array of dim nyears, nbins, nsurvey
- MLpred - predicted mean length - matrix of nyears x nfleet
- MWpred - predicted mean weight - matrix of nyears x nfleet
- CAApred - predicted catch at age - array of nyears, maxage, nfleet
- CALpred - predicted catch at length - array of nyears, nbins, nfleet
- Cpred - predicted catch in weight - matrix of nyears x nfleet
- CN - predicted catch in numbers - matrix of nyears x nfleet
- nll - Total objective function of the model - numeric

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

- 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`.
- report - a list of model output reported from the TMB executable, i.e. `obj$report()`. See Misc.

`data` A list of the data inputs for the RCM.

`config` A data frame describing configuration of the RCM (not currently used).

Author(s)

Q. Huynh

See Also

plot.RCModel RCM

`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 parameters, e.g. R0, steepness, 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)

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

Arguments

`x`	An S4 object of class Assessment of RCModel.
`...`	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 <- SP(Data = swordfish)
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 closed-loop simulation. 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, MP, sim = 1, plot_legend = FALSE)
```

Arguments

`MSE`	An object of class MSE.
`MP`	Character. The name of the management procedure created by `make_MP` containing the assessment model.
`sim`	Integer between 1 and MSE@nsim. The simulation number for which the retrospectives will be plotted.
`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 SSB, depletion, fishing mortality, and vulnerable biomass (VB) estimated in the MP over the course of the closed-loop simulation against the values generated in the operating model (both historical and projected).

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.

Author(s)

Q. Huynh

See Also

diagnostic

Examples

```
SCA_40_10 <- make_MP(SP, HCR_MSY, diagnostic = "full")
OM <- MSEtool::testOM; OM@proyears <- 20
myMSE <- MSEtool::runMSE(OM = OM, MPs = "SCA_40_10")
retrospective_AM(myMSE, MP = "SCA_40_10", sim = 1)
```

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,
  AddInd = "B",
  SR = c("BH", "Ricker"),
  vulnerability = c("logistic", "dome"),
  CAA_dist = c("multinomial", "lognormal"),
  CAA_multiplier = 50,
  rescale = "mean1",
  max_age = Data@MaxAge,
  start = NULL,
  fix_h = TRUE,
  fix_F_equilibrium = TRUE,
  fix_omega = TRUE,
  fix_tau = TRUE,
  LWT = NULL,
  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,
  AddInd = "B",
  SR = c("BH", "Ricker"),
  vulnerability = c("logistic", "dome"),
  CAA_dist = c("multinomial", "lognormal"),
  CAA_multiplier = 50,
  rescale = "mean1",
  max_age = Data@MaxAge,
  start = NULL,
  fix_h = TRUE,
  fix_F_equilibrium = TRUE,
  fix_omega = TRUE,
  fix_tau = TRUE,
  LWT = NULL,
  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,
  AddInd = "B",
  SR = c("BH", "Ricker"),
  vulnerability = c("logistic", "dome"),
  CAA_dist = c("multinomial", "lognormal"),
  CAA_multiplier = 50,
  rescale = "mean1",
  max_age = Data@MaxAge,
  start = NULL,
  fix_h = TRUE,
  fix_U_equilibrium = TRUE,
  fix_tau = TRUE,
  LWT = NULL,
  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
AddInd	A vector of integers or character strings indicating the indices to be used in the model. Integers assign the index to the corresponding index in Data@AddInd, "B" (or 0) represents total biomass in Data@Ind, "VB" represents vulnerable biomass in Data@VInd, and "SSB" represents spawning stock biomass in Data@SpInd. Vulnerability to the survey is fixed in the model.
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.
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, 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, value based on Data@sigmaR.
LWT	A vector of likelihood weights for each survey.

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 diagnostics 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 arguments 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).

In SCA and SCA2, annual F's are estimated parameters assuming continuous fishing over the year, while an annual harvest rate from pulse fishing in the middle of the year is estimated in SCA_Pope.

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, e.g. `CAA_multiplier = 0.1` multiplies the sample size to 10% of the original 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` For all models except `SCA2`, unfished recruitment. Otherwise, 150% Data@OM$R0[x] is used in closed-loop, and 400% of mean catch otherwise.
- `h` Steepness. Otherwise, Data@steep[x] is used, or 0.9 if empty.
- `meanR` For `SCA2`, mean recruitment.
- `vul_par` Vulnerability parameters, see next paragraph.
- `F` A vector of length nyears for year-specific fishing mortality.
- `F_equilibrium` Equilibrium fishing mortality leading into first year of the model (to determine initial depletion). By default, 0.
- `omega` Lognormal SD of the catch (observation error) for all models except `SCA_Pope`. By default, Data@CV_Cat[x].
- `tau` Lognormal SD of the recruitment deviations (process error). By default, Data@sigmaR[x].

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 = MSEtool::SimulatedData)
res2 <- SCA2(Data = MSEtool::SimulatedData)

compare_models(res, res2)
```

SCA_RWM *SCA with random walk in M*

Description

SCA_RWM is a modification of SCA that incorporates a random walk in M in logit space (constant with age). Set the variance to a small value (0.001) in order to fix M for all years, which is functionally equivalent to SCA.

Usage

```
SCA_RWM(
  x = 1,
  Data,
  AddInd = "B",
  SR = c("BH", "Ricker"),
  vulnerability = c("logistic", "dome"),
  CAA_dist = c("multinomial", "lognormal"),
  CAA_multiplier = 50,
  rescale = "mean1",
  max_age = Data@MaxAge,
  start = NULL,
  fix_h = TRUE,
  fix_F_equilibrium = TRUE,
  fix_omega = TRUE,
  fix_tau = TRUE,
  LWT = NULL,
  early_dev = c("comp_onegen", "comp", "all"),
  late_dev = "comp50",
  refyear = expression(length(Data@Year)),
  M_bounds = c(0.5, 2),
  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
AddInd	A vector of integers or character strings indicating the indices to be used in the model. Integers assign the index to the corresponding index in Data@AddInd,

"B" (or 0) represents total biomass in Data@Ind, "VB" represents vulnerable biomass in Data@VInd, and "SSB" represents spawning stock biomass in Data@SpInd.

`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.

`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_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`.

`LWT` A vector of likelihood weights for each survey.

`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.

`refyear` An expression for the year for which M is used to report MSY and depletion reference points. By default, terminal year.

M_bounds	A numeric vector of length 2 to indicate the minimum and maximum M in the random walk as a proportion of the starting M (M_start). The default min and max are 50% and 200%, respectively.
integrate	Logical, whether the likelihood of the model integrates over the likelihood of the recruitment deviations and M random walk (thus, treating it as a random effects/state-space variable). Otherwise, recruitment deviations and the random walk are penalized parameters.
silent	Logical, passed to MakeADFun, whether TMB will print trace information during optimization. Used for diagnostics 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 arguments 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.

Details

The model estimates year-specific M (constant with age) as a random walk in logit space, bounded by a proportion of M_start (specified in M_bounds).

The starting value for the first year M (M_start) is Data@Mort[x] and is fixed. The fixed SD of the random walk (tau_M) is 0.05.

Alternative values can be provided in the start list (see example).

See SCA for all other information about the structure and setup of the model.

The SCA builds in a stock-recruit relationship into the model. Annual unfished and MSY reference points are calculated and reported in TMB_report.

Value

An object of class Assessment.

Author(s)

Q. Huynh

See Also

SCA

Examples

```
res <- SCA_RWM(Data = MSEtool::SimulatedData, start = list(M_start = 0.4, tau_M = 0.05))
res2 <- SCA(Data = MSEtool::SimulatedData)
res3 <- SCA_RWM(Data = MSEtool::SimulatedData, start = list(M_start = 0.4, tau_M = 0.001))


compare_models(res, res2, res3)
```

`Shortcut` *Assessment emulator as a shortcut to model fitting in closed-loop simulation*

Description

Functions (class Assessment) that emulate a stock assessment by sampling the operating model biomass and abundance (with observation error, autocorrelation, and bias) instead of fitting a model. This output can then be passed onto a harvest control rule (HCR function). To utilize the shortcut method in closed-loop simulation, use make_MP with `Shortcut` as the Assessment function. `Perfect` assumes no error in the assessment model and is useful for comparing the behavior of different harvest control rules.

Usage

```
Shortcut(
  x = 1,
  Data,
  method = c("B", "N", "RF"),
  B_err = c(0.3, 0.7, 1),
  N_err = c(0.3, 0.7, 1),
  R_err = c(0.3, 0.7, 1),
  F_err = c(0.3, 0.7, 1),
  ...
)

Perfect(x, 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.
`method`	Indicates where the error in the OM is located. For "B", OM biomass is directly sampled with error. For "N", OM abundance-at-age is sampled and biomass subsequently calculated. For "RF", recruitment and F are sampled to calculate abundance and biomass. There is no error in biological parameters for "N" and "RF".

B_err	If method = "B", a vector of length three that specifies the standard deviation (in logspace), autocorrelation, and bias (1 = unbiased) for biomass.
N_err	Same as B_err, but for abundance when method = "N".
R_err	Same as B_err, but for recruitment when method = "RF".
F_err	Same as B_err, but for fishing mortality when method = "RF".
...	Other arguments (not currently used).

Details

Currently there is no error in FMSY (the target F in the HCR in SAMtool).

See Wiedenmann et al. (2015) for guidance on the magnitude of error for the shortcut emulator.

Value

An object of class Assessment.

Author(s)

Q. Huynh

References

Wiedenmann, J., Wilberg, M.J., Sylvia, A., and Miller, T.J. 2015. Autocorrelated error in stock assessment estimates: Implications for management strategy evaluation. Fisheries Research 172: 325-334.

Examples

```
Shortcut_4010 <- make_MP(Shortcut, HCR40_10)
Shortcut_Nerr <- make_MP(Shortcut, HCR40_10, method = "N", N_err = c(0.1, 0.1, 1)) # Highly precise!


# Compare the shortcut method vs. fitting an SCA model with a 40-10 control rule
MSE <- runMSE(testOM, MPs = c("Shortcut_4010", "SCA_4010"))


# Compare the performance of three HCRs
Perfect_4010 <- make_MP(Perfect, HCR40_10)
Perfect_6020 <- make_MP(Perfect, HCR60_20)
Perfect_8040MSY <- make_MP(Perfect, HCR_ramp, OCP_type = "SSB_SSBMSY", TOCP = 0.8, LOCP = 0.4)


MSE <- runMSE(testOM, MPs = c("Perfect_4010", "Perfect_6020", "Perfect_8040MSY"))
```

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 MSEextra.

Usage

```
SP(
  x = 1,
  Data,
  AddInd = "B",
  rescale = "mean1",
  start = NULL,
  fix_dep = TRUE,
  fix_n = TRUE,
  LWT = NULL,
  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,
  AddInd = "B",
  rescale = "mean1",
  start = NULL,
  fix_dep = TRUE,
  fix_n = TRUE,
  fix_sigma = TRUE,
  fix_tau = TRUE,
  LWT = NULL,
```

```
  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.
AddInd	A vector of integers or character strings indicating the indices to be used in the model. Integers assign the index to the corresponding index in Data@AddInd, "B" (or 0) represents total biomass in Data@Ind, "VB" represents vulnerable biomass in Data@VInd, and "SSB" represents spawning stock biomass in Data@SpInd.
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.
LWT	A vector of likelihood weights for each survey.
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. To override and directly provide the r-prior mean and standard deviation, use the start list, e.g. start = list(r_prior = c(0.1,0.05)) (mean of 0.1 and s.d. of 0.05).
SR_type	If use_r_prior = TRUE, the stock-recruit relationship used to calculate the stock-recruit alpha parameter from steepness and unfished spawners-per-recruit. Used to develop the r prior.
silent	Logical, passed to MakeADFun, whether TMB will print trace information during optimization. Used for diagnostics 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

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

- MSY Maximum sustainable yield.. Otherwise, 300% of mean catch by default.
- FMSY Steepness. Otherwise, Data@Mort[x] or 0.2 is used.
- dep Initial depletion (B/B0) in the first year of the model. By default, 1.
- n The production function exponent that determines BMSY/B0. By default, 2 so that BMSY/B0 = 0.5.
- sigma Lognormal SD of the index (observation error). By default, 0.05. Not used with multiple indices.
- tau Lognormal SD of the biomass deviations (process error) in SP_SS. By default, 0.1.

Multiple indices are supported in the model.

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)


plot(res)
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))


plot(res_SS)


#### 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)
```

SSS *Simple Stock Synthesis*

Description

A simple age-structured model (SCA_Pope) fitted to a time series of catch going back to unfished conditions. Terminal depletion (ratio of current biomass to unfished biomass) is by default fixed to 0.4. Selectivity is fixed to the maturity ogive, although it can be overridden with the start argument. The sole parameter estimated is R0 (unfished recruitment), with no process error.

Usage

```
SSS(
  x = 1,
  Data,
  dep = 0.4,
  SR = c("BH", "Ricker"),
  rescale = "mean1",
  start = NULL,
  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
dep	Depletion value to use in the model. Can be an expression that will be evaluated inside the function.
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 named list of starting values. Entries can be expressions that are evaluated in the function: • R0 Unfished recruitment • vul_par A length-two vector for the age of 95% and 50% fleet selectivity. Fixed to maturity otherwise.
silent	Logical, passed to MakeADFun, whether TMB will print trace information during optimization. Used for diagnostics 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).
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 arguments for optimization to be passed to nlminb.
...	Other arguments to be passed (not currently used).

Details

In SAMtool, SSS is an implementation of SCA_Pope with fixed final depletion (in terms of total biomass, not spawning biomass) assumption.

Value

An object of class Assessment.

Author(s)

Q. Huynh

References

Cope, J.M. 2013. Implementing a statistical catch-at-age model (Stock Synthesis) as a tool for deriving overfishing limits in data-limited situations. Fisheries Research 142:3-14.

Examples

```
res <- SSS(Data = Red_snapper)

SSS_MP <- make_MP(SSS, HCR40_10, dep = 0.3) # Always assume depletion = 0.3
```

Sub_cpars *Rapid Conditioning Model (RCM)*

Description

Intended for conditioning operating models for MSEtool. For data-limited stocks, this function can generate a range of potential depletion scenarios inferred from sparse data. From a historical time series of total catch or effort, and potentially age/length compositions and multiple indices of abundance, the RCM 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 a statistical catch-at-age model (with the predicted catch equal to the observed catch). Alternatively one can do a single model fit and sample the covariance matrix to generate an operating model with uncertainty based on the model fit. 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

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

RCM(OM, data, ...)

## S4 method for signature 'OM,list'
RCM(
  OM,
  data,
  condition = c("catch", "catch2", "effort"),
  selectivity = "logistic",
  s_selectivity = NULL,
  LWT = list(),
  comp_like = c("multinomial", "lognormal"),
  ESS = c(30, 30),
  prior = list(),
  max_F = 3,
  cores = 1L,
  integrate = FALSE,
  mean_fit = FALSE,
  drop_nonconv = FALSE,
  drop_highF = FALSE,
  control = list(iter.max = 2e+05, eval.max = 4e+05),
  ...
)

## S4 method for signature 'OM,Data'
RCM(
  OM,
  data,
  condition = c("catch", "catch2", "effort"),
  selectivity = "logistic",
  s_selectivity = NULL,
  LWT = list(),
  comp_like = c("multinomial", "lognormal"),
  ESS = c(30, 30),
  prior = list(),
  max_F = 3,
  cores = 1L,
  integrate = FALSE,
  mean_fit = FALSE,
  drop_nonconv = FALSE,
  drop_highF = FALSE,
  control = list(iter.max = 2e+05, eval.max = 4e+05),
  ...
)
```

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).
sims	A logical vector of length OM@nsim or a numeric vector indicating which simulations to keep.
data	Data inputs formatted in a list object (preferred). Alternatively, data can be a Data S4 object. See Data section below.
...	Other arguments to pass in for starting values of parameters and fixing parameters. See details.
condition	String to indicate whether the RCM is conditioned on "catch" (where F are estimated parameters), "catch2" (where F is solved internally using Newton's method), or "effort".
selectivity	A character vector of length nfleet to indicate "logistic", "dome", or "free" selectivity for each fleet in Chist. If there is time-varying selectivity, this is a character vector of length nsel_block (see Data section below). "free" indicates independent selectivity parameters for each age, and additional modifications for fixing selectivity parameters will likely be needed. See Additional arguments section.
s_selectivity	A vector of length nsurvey to indicate the selectivity of the corresponding columns in data$Index. Use "B" for total biomass, or "SSB" for spawning biomass (by default, "B" is used). Use numbers if the survey selectivity follows a fleet (corresponding to the columns in data$Chist, e.g., 1 = first fleet/column and so on). If the survey selectivity is otherwise independent of anything else in the model, use "logistic", "dome", or "free" to specify the functional form of selectivity, and see Additional arguments section for setup of survey selectivity parameters. See selectivity vignette for more information.
LWT	A named list of likelihood weights for the RCM. See below.
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 age and length arrays as needed.
prior	A named list (R0, h, M, and q) to provide the mean and standard deviations of prior distributions for those parameters. R0 and M priors lognormal (mean in normal space, SD in lognormal space). Beverton-Holt steepness uses a beta prior, while survey q and Ricker steepness use normal priors. For survey q, provide a matrix for nsurvey rows and 2 columns (for mean and SD). For all others, provide a length-2 vector for the mean and SD. See vignette for full description.

max_F	The maximum F for any fleet in the scoping model (higher F's in the model are penalized in the objective function). See also drop_highF.
cores	Integer for the number of CPU cores for the stock reduction analysis.
integrate	Logical, whether to treat recruitment deviations as penalized parameters in the likelihood (FALSE) or random effects to be marginalized out of the likelihood (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 RCM, including fits where F = NA.
drop_highF	Logical, whether to drop fits of the RCM where F = max_F.
control	A named list of arguments (e.g, max. iterations, etc.) for optimization, to be passed to the control argument of nlminb.

Details

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

Survey selectivity is estimable 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 s_selectivity).

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

If the operating model OM uses time-varying growth or M, then those trends will be used in the RCM as well. Non-stationary productivity creates ambiguity in the calculation and interpretation of depletion and MSY reference points.

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

To play with alternative fits by excluding indices, for example, or other optional data, set the corresponding likelihood weight to zero. The model will still generate the inferred index but the data won't enter the likelihood. See section on likelihood weights.

Value

An object of class RCModel (see link for description of output).

Vignette

Three vignettes are available for the RCM:

- General overview of approach
- Mathematical description
- Setup of selectivity settings (useful for more data-rich cases)

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` variable can be a named list that includes:

- 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`).
- C_sd - A vector or matrix of standard deviations (lognormal distribution) for the catches in `Chist`. If not provided, the default is 0.01. Only used if `condition = "catch"`.
- 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 - Obsolete as of version 2.0. See `s_selectivity` argument.
- CAA - Fishery age composition matrix with nyears rows and OM@maxage+1 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.
- MS - A vector of fishery mean size (MS, either mean length or mean weight) observations (length OM@nyears), or if multiple fleets: matrix of dimension: nyears and nfleets. Generally, mean lengths should not be used if `CAL` is also provided, unless mean length and length comps are independently sampled.
- MS_type - A character (either `"length"` (default) or `"weight"`) to denote the type of mean size data.
- MS_cv - The coefficient of variation of the observed mean size. If there are multiple fleets, a vector of length nfleet. Default is 0.2.
- s_CAA - Survey age composition data, an array of dimension nyears, maxage+1, 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. Alternatively, if one has a full CAA matrix, one could instead estimate "artificial" rec devs to generate the initial numbers-at-age (and hence initial depletion) in the first year of the model (see additional arguments).
- C_eq_sd - A vector of standard deviations (lognormal distribution) for the equilibrium catches in `C_eq`. If not provided, the default is 0.01. Only used if `condition = "catch"`.

- 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_units - Optional, an integer vector to indicate whether indices are biomass based (1) or abundance-based (0). By default, all are biomass-based.
- age_error - Optional, a square matrix of maxage + 1 rows and columns to specify ageing error. The aa-th column assigns a proportion of the true age in the a-th row to observed age. Thus, all rows should sum to 1. Default is an identity matrix (no ageing error).
- sel_block - Optional, for time-varying fleet selectivity (in time blocks), a integer matrix of nyears rows and nfleet columns to assigns a selectivity function to a fleet for certain years. See the selectivity vignette for more details.

Alternatively, the `data` input can be a Data S4 object which will retrieve data from the following slots:

- Data@Cat - catch series (single fleet with the Data S4 object)
- Data@Effort - effort series
- Data@CAA - fishery age composition
- Data@CAL, Data@CAL_mids - fishery length composition and corresponding length bins
- Data@Ind, Data@SpInd, Data@VInd, Data@AddInd - indices of abundance
- Data@CV_Ind, Data@CV_SpInd, Data@CV_VInd, Data@CV_AddInd - annual coefficients of variation for the corresponding indices of abundance. CVs will be converted to lognormal standard deviations.
- Data@ML - fishery mean lengths
- Data@AddIndV, Data@AddIndType, Data@AddIunits - Additional information for indices in Data@AddInd: selectivity and units (i.e., biomass or abundance).

There is no slot in the Data S4 object for the equilibrium catch/effort. These can be passed in the function call, i.e., `RCM(OM,Data,C_eq = C_eq,...)`.

Additional arguments

For `RCM`, 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 LFS (length of full selectivity), L5 (length of 5 percent selectivity), and Vmaxlen (selectivity at length Linf). By default, the starting values are values from the OM object. If any selectivity = "free", then this matrix needs to be of maxage rows where the row specifies the selectivity at age. See the selectivity vignette for more information.
- s_vul_par: A matrix of 3 rows and nsurvey columns for starting values for fleet selectivity. Same setup as vul_par. These values are only used if `s_selectivity = "est"` for the corresponding fleet. Otherwise, placeholders should be used to complete the matrix.

- map_vul_par: An integer matrix of the same dimension as vul_par. This is the 'map' argument for vul_par in TMB, see MakeADFun, which indicates whether selectivity parameters are fixed or estimated. 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. Unused cells in the vul_par matrix should be given NA in the map matrix.
- map_s_vul_par: The map argument for the survey selectivity parameters (same dimension as s_vul_par). Placeholder parameters should have a map value of NA.
- map_log_early_rec_dev: A vector of length OM@maxage - 1 that indexes which recruitment deviates for the cohorts in the first year of the model are fixed (using NA) or estimated (a separate integer). By default, no deviates are estimated.
- map_log_rec_dev: A vector of length OM@nyears that indexes which recruitment deviates are fixed (using NA) or estimated (a separate integer). By default, all deviates are estimated.
- plusgroup: Logical for whether the maximum age is a plusgroup or not. By default, TRUE.
- fix_dome: Logical for whether the dome selectivity parameter for fleets is fixed. Used primarily for backwards compatibility, this is overridden by map_vul_par.
- resample: Logical, whether the OM conditioning parameters (recruitment, fishing mortality, SSB, selectivity, etc.) are obtained by sampling the Hessian matrix from a single model fit. By default FALSE. This feature requires identical biological parameters among simulations.

Likelihood weights

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

- Chist, CAA, CAL, MS, C_eq: A vector of length nfleet for each.
- Index, 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 will be equal to the presumed multinomial sample size. Argument ESS provides a shortcut to cap the multinomial sample size for age and length comps.

Author(s)

Q. Huynh

See Also

plot.RCModel RCModel

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 = MSEtool::SimulatedData)
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)
```

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

userguide *Get the SAMtool vignettes*

Description

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

Usage

```
userguide()
```

Value

Displays a browser webpage of URL links to package vignettes.

Examples

```
userguide()
```

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 and per-recruit quantities are then calculated from the VPA output.

Usage

```
VPA(
  x = 1,
  Data,
  AddInd = "B",
  expanded = FALSE,
  SR = c("BH", "Ricker"),
  vulnerability = c("logistic", "dome", "free"),
  start = list(),
  fix_h = TRUE,
  fix_Fratio = TRUE,
  fix_Fterm = FALSE,
  LWT = NULL,
  shrinkage = list(),
  nitF = 5L,
  min_age = "auto",
  max_age = "auto",
  refpt = list(),
  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
AddInd	A vector of integers or character strings indicating the indices to be used in the model. Integers assign the index to the corresponding index in Data@AddInd, "B" (or 0) represents total biomass in Data@Ind, "VB" represents vulnerable biomass in Data@VInd, and "SSB" represents spawning stock biomass in Data@SpInd.
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.
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 MSY and unfished reference points.
fix_Fratio	Logical, whether the ratio of F of the plus-group to the previous age class is fixed in the model.
fix_Fterm	Logical, whether to fix the value of the terminal F.
LWT	A vector of likelihood weights for each survey.
shrinkage	A named list of up to length 2 to constrain parameters: • vul - a length two vector that constrains the vulnerability-at-age in the most recent years. The first number is the number of years in which vulnerability will be constrained (as a random walk in log space), the second number is the standard deviation of the random walk. The default • R - a length two vector that constrains the recruitment estimates in the most recent years. The first number is the number of years in which recruitment will be constrained (as a random walk in log space), 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).
min_age	An integer to specify the smallest age class in the VPA. By default, the youngest age with non-zero CAA in the terminal year is used.
max_age	An integer to specify the oldest age class in the VPA. By default, the oldest age with non-zero CAA for all years is used.
refpt	A named list of how many years to average parameters for calculating reference points, yield per recruit, and spawning potential ratio: • vul An integer for the number of most recent years to average the vulnerability schedule (default is 3).

- `R` A length two for the quantile used to calculate recruitment in the year following the terminal year and the number of years from which that quantile is used, i.e., `c(0.5,5)` is the default that calculates median recruitment from the most recent 5 years of the model.

`silent` Logical, passed to `MakeADFun`, whether TMB will print trace information during optimization. Used for diagnostics 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 arguments 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 `Fterm` is the apical terminal F if a functional form for vulnerability is used in the terminal year, i.e., when `vulnerability = "logistic"` or `"free"`. If the terminal F-at-age are otherwise independent parameters, `Fterm` 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:

- `Fterm` The terminal year fishing mortality. This is the apical F when `vulnerability = "logistic"` or `"free"`.
- `Fratio` 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.

MSY and depletion reference points are calculated by fitting the stock recruit relationship to the recruitment and SSB estimates. Per-recruit quantities are also calculated, which may be used in harvest control rules.

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