Package: wham (via r-universe)

September 20, 2024

Title Woods Hole Assessment Model (WHAM)

Version 1.0.9

Description The Woods Hole Assessment Model (WHAM) is a state-space age-structured stock assessment model that can include environmental effects on population processes. WHAM can be configured to estimate a range of assessment models: a traditional statistical catch-at-age (SCAA) model with recruitments as fixed effects, SCAA with recruitments as random effects, or a ``full state-space'' model with abundance at all ages treated as random effects. WHAM is a generalization of the R and TMB code from Miller et al. (2016), Miller and Hyun (2018), and Miller et al. (2018). WHAM also has many similarities of input data sources with ASAP (Legault and Restrepo 1999) and provides functions to generate a WHAM input file from an ASAP3 dat file. Many of the plotting functions for input data, results, and diagnostics are modified from ASAP code written by Chris Legault and Liz Brooks.

Depends R (>= 3.6.0)

Imports TMB (>= 1.7.20), plotrix (>= 3.7-5), ellipse (>= 0.4.1), Hmisc (>= 4.4-1), gplots (>= 3.0.1.1), fields (>= 9.6), RColorBrewer (>= 1.1-2), colorspace (>= 1.4-1), mnormt (>= 1.5-5), Deriv (>= 3.8.5), tidyr (>= 1.1.2), dplyr (>= 1.0.2), ggplot2 (>= 3.3.2), viridis (>= 0.5.1), sessioninfo (>= 1.1.1), abind (>= 1.4-5), rmarkdown (>= 2.11), knitr (>= 1.37), pander (>= 0.6.4), kableExtra (>= 1.3.4)

Remotes kaskr/adcomp/TMB

LinkingTo TMB, RcppEigen

URL `https://timjmiller.github.io/wham/`,
`https://github.com/timjmiller/wham`

BugReports `https://github.com/timjmiller/wham/issues`

License GPL-3

LazyData true

VignetteBuilder knitr

RoxygenNote 7.3.0

Encoding UTF-8

Suggests testthat

Repository https://noaa-fisheries-integrated-toolbox.r-universe.dev

RemoteUrl https://github.com/timjmiller/wham

RemoteRef HEAD

RemoteSha 89b3fd5c6252177d224c9fd2e8364da1716821ea

Contents

check_convergence *Check convergence of WHAM model*

Description

Access quick convergence checks from ‘TMB‘ and ‘nlminb‘.

Usage

```
check_convergence(mod, ret = FALSE, f = "")
```

Arguments

mod	output from fit_wham
ret	T/F, return list? Default = FALSE, just prints to console

Value

a list with at least the first three of these components:

$convergence From stats::nlminb, "0 indicates successful convergence for nlminb"

$maxgr Max absolute gradient value, from ‘max(abs(mod$gr(mod$opt$par)))‘

$maxgr_par Name of parameter with max gradient

$is_sdrep If TMB::sdreport was performed for this model, this indicates whether it performed without error

$na_sdrep If TMB::sdreport was performed without error for this model, this indicates which (if any) components of the diagonal of the inverted hessian were returned as NA

See Also

fit_wham, fit_tmb, stats::nlminb

Examples

```
## Not run:
data("input4_SNEMAYT") # load SNEMA yellowtail flounder data and parameter settings
mod = fit_wham(input4_SNEMAYT) # using default values
check_convergence(mod)

## End(Not run)
```

check_estimability *Check for identifiability of fixed effects Originally provided by https://github.com/kaskr/TMB_contrib_R/TMBhelper Internal function called by fit_tmb.*

Description

check_estimability calculates the matrix of second-derivatives of the marginal likelihood w.r.t. fixed effects, to see if any linear combinations are not estimable (i.e. cannot be uniquely estimated conditional upon model structure and available data, e.g., resulting in a likelihood ridge and singular, non-invertable Hessian matrix)

Usage

```
check_estimability(obj, h)
```

Arguments

`obj`	The compiled object
`h`	optional argument containing pre-computed Hessian matrix

Value

A tagged list of the hessian and the message

`compare_wham_models` *Compare multiple WHAM (or ASAP) models*

Description

After fitting multiple WHAM (or ASAP) models, `compare_wham_models` produces plots and a table of AIC and Mohn's rho to aid model comparison.

Usage

```
compare_wham_models(
  mods,
  do.table = TRUE,
  do.plot = TRUE,
  fdir = getwd(),
  table.opts = NULL,
  plot.opts = NULL,
  fname = NULL,
  sort = NULL,
  calc.rho = NULL,
  calc.aic = NULL,
  do.print = NULL
)
```

Arguments

`mods` (named) list of fit WHAM/ASAP models. To read in ASAP model output, use `read_asap3_fit`. If no names are given, 'm1', 'm2', ... will be used.

`do.table` T/F, produce table of AIC and/or Mohn's rho? Default = TRUE.

`do.plot` T/F, produce plots? Default = TRUE.

`fdir` character, path to directory to save table and/or plots. Default = `getwd()`.

`table.opts` list of options for AIC/rho table:

- `$fname` character, filename to save CSV results table (.csv will be appended). Default = `'model_comparison'`.
- `$sort` T/F, sort by AIC? Default = TRUE.
- `$calc.rho` T/F, calculate Mohn's rho? Retrospective analysis must have been run for all modes. Default = TRUE.

`$calc.aic` T/F, calculate AIC? Default = TRUE.

`$print` T/F, print table to console? Default = TRUE.

`$save.csv` T/F, save table as a CSV file? Default = TRUE.

`plot.opts` list of options for plots:

`$out.type` character, either `'pdf'` or `'png'` (default = `'png'` because I am not sure `system('pdftk')` will work across platforms.)

`$ci` vector of T/F, length = 1 (applied to all models) or number of models

`$years` vector, which years to plot? Default = all (model and projection years).

`$which` vector, which plots to make? Default = all. See details.

`$relative.to` character, name of "base" model to plot differences relative to.

`$alpha` scalar, (1-alpha)% confidence intervals will be plotted. Default = 0.05 for 95% CI.

`$ages.lab` vector, overwrite model age labels.

`$kobe.yr` integer, which year to use in Kobe plot (relative status). Default = terminal model year.

`$M.age` integer, which age to use in M time-series plot. Default = `max(data$which_F_age)` (max age of F to use for full total F).

`$return.ggplot` T/F, return a list of ggplot2 objects for later modification? Default = TRUE.

`$kobe.prob` T/F, print probabilities for each model in each quadrant of Kobe plot? Default = TRUE.

`$refpt` "XSPR" or "MSY", which reference point to use. Default = "XSPR".

Details

`plot.opts$which` specifies which plots to make:

1 3-panel of SSB (spawning stock biomass), F (fully-selected fishing mortality), and Recruitment

2 CV (coefficient of variation) for SSB, F, and Recruitment

3 Fleet selectivity (by block, averaged across years)

4 Index selectivity (by block, averaged across years)

5 Selectivity tile (fleets + indices, useful for time-varying random effects)

6 M time series (natural mortality, can specify which age with plot.opts$M.age)

7 M tile (useful for time-varying random effects)

8 3-panel of F X% SPR, SSB at F_X%SPR, and yield at F_X%SPR

9 2-panel of relative status (SSB / SSB at F_X%SPR and F / F_X%SPR)

10 Kobe status (relative SSB vs. relative F)

If `plot.opts$return.ggplot` = `TRUE`, a list `g` is returned holding the above ggplot2 objects for later modification. `g[[i]]` holds the plot corresponding to `i` above, e.g. `g[[2]]` is the CV plot.

Value

a list with the following components:

- `daic` Vector of delta-AIC by model (if `do.table=T` and `table.opts$calc.aic=T`)
- `aic` Vector of AIC by model (if `do.table=T` and `table.opts$calc.aic=T`)
- `rho` Matrix of Mohn's rho by model (if `do.table=T` and `table.opts$calc.rho=T`)
- `best` Name of best model (lowest AIC) (if `do.table=T` and `table.opts$calc.aic=T`)
- `tab` Results table of AIC and Mohn's rho (if `do.table=T`)
- `g` List of ggplot2 objects for later modification (if `do.plot=T` and `plot.opts$return.ggplot=T`)

See Also

`fit_wham`, `read_asap3_fit`, `read_wham_fit`

Examples

```
## Not run:
base <- read_asap3_fit()
m1 <- fit_wham(input1)
m2 <- fit_wham(input2)
mods <- list(base=base, m1=m1, m2=m2)
res <- compare_wham_models(mods)

## End(Not run)
```

---

extract_fixed *Extract fixed effects Originally provided by https://github.com/kaskr/TMB_contrib_R/TMBhelper Internal function called by* `check_estimability`.

Description

`extract_fixed` extracts the best previous value of fixed effects, in a way that works for both mixed and fixed effect models

Usage

```
extract_fixed(obj)
```

Arguments

`obj`	The compiled object

Value

A vector of fixed-effect estimates

fit_peel *Fit model peeling off* i *years of data*

Description

Internal function called by retro for *i* in 1–n.peels. Fits the model peeling off *i* years of data (calls fit_tmb).

Usage

```
fit_peel(
  peel,
  input,
  do.sdrep = FALSE,
  n.newton = 3,
  MakeADFun.silent = FALSE,
  retro.silent = FALSE,
  save.input = FALSE
)
```

Arguments

peel	Integer, number of years of data to remove before model fitting.
input	input with same structure as that provided by prepare_wham_input. May want to use input$par = model$parList to start at MLEs.
do.sdrep	T/F, calculate standard deviations of model parameters? Default = FALSE.
n.newton	integer, number of additional Newton steps after optimizafit_tmbtion for each peel. Default = 3.
MakeADFun.silent	T/F, Passed to silent argument of TMB::MakeADFun. Default = FALSE.
retro.silent	T/F, Passed to argument of internal fit_peel function. Determines whether peel number is printed to screen. Default = FALSE.
save.input	T/F, should modified input list be saved? Necessary to project from a peel but increases model object size. Default = FALSE.

Value

out, output of fit_tmb for peel *i*

See Also

fit_wham, retro, fit_tmb

fit_tmb *Fit TMB model using nlminb*

Description

Runs optimization on the TMB model using stats::nlminb. If specified, takes additional Newton steps and calculates standard deviations. Internal function called by fit_wham.

Usage

```
fit_tmb(
  model,
  n.newton = 3,
  do.sdrep = TRUE,
  do.check = FALSE,
  save.sdrep = FALSE
)
```

Arguments

model	Output from TMB::MakeADFun.
n.newton	Integer, number of additional Newton steps after optimization. Default = 3.
do.sdrep	T/F, calculate standard deviations of model parameters? See TMB::sdreport. Default = TRUE.
do.check	T/F, check if model parameters are identifiable? Runs internal check_estimability, originally provided by https://github.com/kaskr/TMB_contrib_R/TMBhelper. Default = TRUE.
save.sdrep	T/F, save the full TMB::sdreport object? If FALSE, only save summary.sdreport) to reduce model object file size. Default = FALSE.

Value

model, appends the following:

model$opt Output from stats::nlminb

model$date System date

model$dir Current working directory

model$rep model$report()

model$TMB_version Version of TMB installed

model$parList List of parameters, model$env$parList()

model$final_gradient Final gradient, model$gr()

model$sdrep Estimated standard deviations for model parameters, TMB::sdreport or summary.sdreport)

See Also

fit_wham, retro, TMBhelper::check_estimability

fit_wham *Fit WHAM model*

Description

Fits the compiled WHAM model using `TMB::MakeADFun` and `stats::nlminb`. Runs retrospective analysis if specified.

Usage

```
fit_wham(
  input,
  n.newton = 3,
  do.sdrep = TRUE,
  do.retro = TRUE,
  n.peels = 7,
  do.osa = TRUE,
  osa.opts = list(method = "cdf", parallel = TRUE),
  do.post.samp = TRUE,
  model = NULL,
  do.check = FALSE,
  MakeADFun.silent = FALSE,
  retro.silent = FALSE,
  do.proj = FALSE,
 proj.opts = list(n.yrs = 3, use.last.F = TRUE, use.avg.F = FALSE, use.FXSPR = FALSE,
   proj.F = NULL, proj.catch = NULL, avg.yrs = NULL, cont.ecov = TRUE, use.last.ecov =
   FALSE, avg.ecov.yrs = NULL, proj.ecov = NULL, cont.Mre = NULL, avg.rec.yrs = NULL,
    percentFXSPR = 100),
  do.fit = TRUE,
  save.sdrep = TRUE
)
```

Arguments

`input` Named list with components:

`$data` Data to fit the assessment model to.

`$par` Parameters, a list of all parameter objects required by the user template (both random and fixed effects). See `MakeADFun`.

`$map` Map, a mechanism for collecting and fixing parameters. See `MakeADFun`.

`$random` Character vector defining the parameters to treat as random effect. See `MakeADFun`.

`$years` Numeric vector of the years which the model spans. Not important for model fitting, but useful for plotting.

`$model_name` Character, name of the model, e.g. `"Yellowtail flounder"`

`$ages.lab` Character vector of the age labels, e.g. `c("1","2","3","4+")`.

`n.newton` integer, number of additional Newton steps after optimization. Passed to `fit_tmb`. Default = `3`.

`do.sdrep` T/F, calculate standard deviations of model parameters? See `sdreport`. Default = `TRUE`.

`do.retro` T/F, do retrospective analysis? Default = `TRUE`.

`n.peels` integer, number of peels to use in retrospective analysis. Default = `7`.

`do.osa` T/F, calculate one-step-ahead (OSA) residuals? Default = `TRUE`. See details. Returned as `mod$osa$residual`.

`osa.opts` list of 2 options (method, parallel) for calculating OSA residuals, passed to `TMB::oneStepPredict`. Default: `osa.opts = list(method="oneStepGaussianOffMode", parallel=TRUE)`. See `make_osa_residuals`.

`do.post.samp` T/F, obtain sample from posterior of random effects? Default = `TRUE`. NOT YET IMPLEMENTED.

`model` (optional), a previously fit wham model.

`do.check` T/F, check if model parameters are identifiable? Passed to `fit_tmb`. Runs internal function `check_estimability`, originally provided by https://github.com/kaskr/TMB_contrib_R/TM Default = `TRUE`.

`MakeADFun.silent`

T/F, Passed to silent argument of `TMB::MakeADFun`. Default = `FALSE`.

`retro.silent` T/F, Passed to argument of internal retro function. Determines whether peel number is printed to screen. Default = `FALSE`.

`do.proj` T/F, do projections? Default = `FALSE`. If true, runs `project_wham`.

`proj.opts` a named list with the following components:

- `$n.yrs` (integer), number of years to project/forecast. Default = `3`.
- `$use.last.F` (T/F), use terminal year F for projections. Default = `TRUE`.
- `$use.FXSPR` (T/F), calculate and use F at X
- `$use.FMSY` (T/F), calculate and use FMSY for projections.
- `$proj.F` (vector), user-specified fishing mortality for projections. Length must equal `n.yrs`.
- `$proj.catch` (vector), user-specified aggregate catch for projections. Length must equal `n.yrs`.
- `$avg.yrs` (vector), specify which years to average over for calculating reference points. Default = last 5 model years, `tail(model$years, 5)`.
- `$cont.ecov` (T/F), continue ecov process (e.g. random walk or AR1) for projections. Default = `TRUE`.
- `$use.last.ecov` (T/F), use terminal year ecov for projections.
- `$avg.ecov.yrs` (vector), specify which years to average over the environmental covariate(s) for projections.
- `$proj.ecov` (vector), user-specified environmental covariate(s) for projections. Length must equal `n.yrs`.
- `$cont.Mre` (T/F), continue M random effects (i.e. AR1_y or 2D AR1) for projections. Default = `TRUE`. If `FALSE`, M will be averaged over `$avg.yrs` (which defaults to last 5 model years).

- `$avg.rec.yrs` (vector), specify which years to calculate the CDF of recruitment for use in projections. Default = all model years.
- `$percentFXSPR` (scalar), percent of F_XSPR to use for calculating catch in projections, only used if proj.opts$use.FXSPR = TRUE. For example, GOM cod uses F = 75
- `$percentFMSY` (scalar), percent of F_MSY to use for calculating catch in projections, only used if $use.FMSY = TRUE.

`do.fit` T/F, fit the model using `fit_tmb`. Default = `TRUE`.

`save.sdrep` T/F, save the full `TMB::sdreport` object? If `FALSE`, only save `summary.sdreport` to reduce model object file size. Default = `TRUE`.

Details

Standard residuals are not appropriate for models with random effects. Instead, one-step-ahead (OSA) residuals can be used for evaluating model goodness-of-fit (Thygeson et al. (2017), implemented in `TMB::oneStepPredict`). Additional OSA residual options are passed to `TMB::oneStepPredict` in a list `osa.opts`. For example, to use the (much faster, ~1 sec instead of 2 min) full Gaussian approximation instead of the (default) generic method, you can use `osa.opts=list(method="fullGaussian")`.

Value

a fit TMB model with additional output if specified:

`$rep` List of derived quantity estimates (see examples)

`$sdrep` Parameter estimates (and standard errors if `do.sdrep=TRUE`)

`$peels` Retrospective analysis (if `do.retro=TRUE`)

`$osa` One-step-ahead residuals (if `do.osa=TRUE`)

See Also

`fit_tmb`, `retro`, `TMB::oneStepPredict`, `project_wham`

Examples

```
## Not run:
data("input4_SNEMAYT") # load SNEMA yellowtail flounder data and parameter settings
mod = fit_wham(input4_SNEMAYT) # using default values
mod = fit_wham(input4_SNEMAYT, do.retro=FALSE, osa.opts=list(method="oneStepGeneric")) # slower OSA method.

names(mod$rep) # list of derived quantities
mod$rep$SSB # get SSB estimates (weight, not numbers)
m1$rep$NAA[,1] # get recruitment estimates (numbers, first column of numbers-at-age matrix)
m1$rep$F[,1] # get F estimates for fleet 1

## End(Not run)
```

make_osa_residuals *Calculate one-step-ahead residuals*

Description

Standard residuals are not appropriate for models with random effects. Instead, one-step-ahead (OSA) residuals can be used for evaluating model goodness-of-fit (Thygeson et al. (2017), implemented in `TMB::oneStepPredict`). OSA residual options are passed to `TMB::oneStepPredict` in a list `osa.opts`. Current options are method: oneStepGaussianOffMode (default), oneStepGaussian, or oneStepGeneric, and parallel: TRUE/FALSE. See `TMB::oneStepPredict` for further details. It is not recommended to run this function (or `TMB::oneStepPredict`) with any random effects and mvtweedie age composition likelihoods due to extensive computational demand. An error will be thrown in such cases. See Trijoulet et al. (2023) for OSA methods for age composition OSA residuals.

Usage

```
make_osa_residuals(
  model,
  osa.opts = list(method = "oneStepGaussianOffMode", parallel = TRUE)
)
```

Arguments

`model`	A fit WHAM model, output from `fit_wham`.

Value

the same fit TMB model with additional elements for osa residuals:

`$OSA.Ecov` data.frame returned by `TMB::oneStepPredict` for environmental observations, if applicable.

`$OSA.agregate` data.frame returned by `TMB::oneStepPredict` for aggregate catch and index observations conditional on any environmental observations, if applicable.

`$OSA.agecomp` data.frame returned by `TMB::oneStepPredict` for age composition observations conditional on any aggregate catch or index, or environmental observations, if applicable.

`$osa` One-step-ahead residuals (if `do.osa=TRUE`)

See Also

`fit_wham`

Examples

```
## Not run:
data("input4_SNEMAYT") # load SNEMA yellowtail flounder data and parameter settings
mod <- fit_wham(input4_SNEMAYT, do.osa =FALSE, do.retro =FALSE)
mod <- make_osa_residuals(mod) # calculate Mohn's rho
plot_wham_output(mod)

## End(Not run)
```

---

mohns_rho *Calculate Mohn's rho for a WHAM model with peels*

Description

Calculate Mohn's rho for a WHAM model with peels

Usage

```
mohns_rho(model)
```

Arguments

`model`	A fit WHAM model, output from `fit_wham` with `do.retro` = `TRUE`.

Value

rho, a vector of Mohn's rho

See Also

`fit_wham`, `retro`

Examples

```
## Not run:
data("input4_SNEMAYT") # load SNEMA yellowtail flounder data and parameter settings
mod = fit_wham(input4_SNEMAYT) # using default values: do.retro = T, n.peels = 7
mohns_rho(mod) # calculate Mohn's rho

## End(Not run)
```

plot_wham_output *Plot WHAM output*

Description

Generates many output plots and tables for a fit WHAM model.

Usage

```
plot_wham_output(
  mod,
  dir.main = getwd(),
  out.type = "png",
  res = 72,
  plot.opts = NULL
)
```

Arguments

mod	output from fit_wham
dir.main	character, directory to save plots to (default = getwd())
out.type	character, either 'html', 'pdf', or 'png' (default = 'png')
res	resolution to save .png files (dpi)
plot.opts	(optional) list of plot modifications

Details

out.type = 'pdf' makes one pdf file of all plots. out.type = 'png' (default) creates a subdirectory ‘plots_png“ in dir.main and saves .png files within. out.type = 'html' makes a html files for viewing plot .png files and html tables of parameter estimates in a browser. out.type = 'pdf' or 'png' makes LaTeX and pdf files of tables of parameter estimates. (tabs: ’input data’, ’diagnostics’, ’results’, ’ref_points’, ’retro’, and ’misc’).

plot.opts holds optional arguments to modify plots:

$ages.lab Character vector, will change age labels in plots (default is 1:n.ages).

$font.family Font family, e.g. "Times".

Plot functions are located in wham_plots_tables.R Table function is located in par_tables_fn.R

See Also

fit_wham, wham_html, wham_plots_tables

Examples

```
## Not run:
data("input4_SNEMAYT") # load fit wham model
mod <- fit_wham(input4_SNEMAYT)
plot_wham_output(mod)

## End(Not run)
```

`prepare_projection` *Prepare input data and parameters to project an already fit WHAM model*

Description

`prepare_projection` is an internal function called by `project_wham`, which in turn is called by `fit_wham` if `do.proj` = `TRUE`.

Usage

```
prepare_projection(model, proj.opts)
```

Arguments

`model` a previously fit wham model

`proj.opts` a named list with the following components:

- `$n.yrs` (integer), number of years to project/forecast. Default = `3`.
- `$use.last.F` (T/F), use terminal year F for projections. Default = `TRUE`.
- `$use.avg.F` (T/F), use average of F over certain years for projections. Default = `FALSE`. Years to average over determined by $avg.yrs defined below.
- `$use.FXSPR` (T/F), calculate and use F at X% SPR for projections. Default = `FALSE`.
- `$use.FMSY` (T/F), calculate and use FMSY for projections. Default = `FALSE`.
- `$proj.F` (vector), user-specified fishing mortality for projections. Length must equal `n.yrs`.
- `$proj.catch` (vector), user-specified aggregate catch for projections. Length must equal `n.yrs`.
- `$avg.yrs` (vector), specify which years to average over for calculating reference points. Default = last 5 model years, `tail(model$years, 5)`.
- `$cont.ecov` (T/F), continue ecov process (e.g. random walk or AR1) for projections. Default = `TRUE`.
- `$use.last.ecov` (T/F), use terminal year ecov for projections.
- `$avg.ecov.yrs` (vector), specify which years to average over the environmental covariate(s) for projections.
- `$proj.ecov` (matrix), user-specified environmental covariate(s) for projections. `n.yrs` x `n.ecov`.

- $cont.Mre (T/F), continue M random effects (i.e. AR1_y or 2D AR1) for projections. Default = TRUE. If FALSE, M will be averaged over $avg.yrs (which defaults to last 5 model years).
- $avg.rec.yrs (vector), specify which years to calculate the CDF of recruitment for use in projections. Default = all model years. Only used when recruitment is estimated as fixed effects (SCAA).
- $percentFXSPR (scalar), percent of F_XSPR to use for calculating catch in projections, only used if $use.FXSPR = TRUE. For example, GOM cod uses F = 75% F_40%SPR, so proj.opts$percentFXSPR = 75. Default = 100.
- $percentFMSY (scalar), percent of F_MSY to use for calculating catch in projections, only used if $use.FMSY = TRUE.
- $proj_F_opt (vector), integers specifying how to configure each year of the projection: 1: use terminal F, 2: use average F, 3: use F at X% SPR, 4: use specified F, 5: use specified catch, 6: use Fmsy. Overrides any of the above specifications.
- $proj_Fcatch (vector), catch or F values to use each projection year: values are not used when using Fmsy, FXSPR, terminal F or average F. Overrides any of the above specifications of proj.F or proj.catch.
- $proj_mature (matrix), user-supplied maturity values for the projection years with dimensions (n.yrs x n_ages).
- $proj_waa (3-d array), user-supplied waa values for the projection years with first and third dimensions equal to that of model$input$data$waa (waa source x n.yrs x n_ages).
- $proj_R_opt (integer), 1: continue any RE processes for recruitment, 2: make projected recruitment consistent with average recruitment in SPR reference points and cancel any bias correction for NAA in projection years.

Value

same as prepare_wham_input, a list ready for fit_wham:

data Named list of data, passed to TMB::MakeADFun

par Named list of parameters, passed to TMB::MakeADFun

map Named list of factors that determine which parameters are estimated, passed to TMB::MakeADFun

random Character vector of parameters to treat as random effects, passed to TMB::MakeADFun

years Numeric vector of representing (non-projection) model years of WHAM model

years_full Numeric vector of representing all model and projection years of WHAM model

ages.lab Character vector of age labels, ending with plus-group

model_name Character, name of stock/model (specified in call to prepare_wham_input)

See Also

prepare_wham_input, project_wham

prepare_wham_input *Prepare input data and parameters for WHAM model*

Description

After the data file is read into R by read_asap3_dat, this function prepares the data and parameter settings for fit_wham. By default, this will set up a SCAA version like ASAP. As of version 1.0.5, if an asap3 object is not provided, a dummy input is generated with some arbitrary assumptions. The various options described below, such as NAA_re and selectivity, can still be used without the asap3 object.

Usage

```
prepare_wham_input(
  asap3 = NULL,
  model_name = "WHAM for unnamed stock",
  recruit_model = 2,
  ecov = NULL,
  selectivity = NULL,
  M = NULL,
  NAA_re = NULL,
  catchability = NULL,
  age_comp = NULL,
  basic_info = NULL
)
```

Arguments

asap3	(optional) list containing data and parameters (output from read_asap3_dat)
model_name	character, name of stock/model
recruit_model	numeric, option to specify stock-recruit model (see details)
ecov	(optional) named list of environmental covariate data and parameters (see details)
selectivity	(optional) list specifying selectivity options by block: models, initial values, parameters to fix, and random effects (see details)
M	(optional) list specifying natural mortality options: model, random effects, initial values, and parameters to fix (see details)
NAA_re	(optional) list specifying options for random effect on numbers-at-age, initial numbers at age, and recruitment model (see details)
catchability	(optional) list specifying options for priors and random effects on catchability (see details)
age_comp	(optional) character or named list, specifies age composition model for fleet(s) and indices (see details)
basic_info	(optional) list of basic population information for use when asap3 is not provided (see details)

Details

recruit_model specifies the stock-recruit model. See wham.cpp for implementation.

= 1 SCAA (without NAA_re option specified) or Random walk (if NAA_re$sigma specified), i.e. predicted recruitment in year i = recruitment in year i-1

= 2 (default) Random about mean, i.e. steepness = 1

= 3 Beverton-Holt

= 4 Ricker

Note: we allow fitting a SCAA model (NAA_re = NULL), which estimates recruitment in every year as separate fixed effect parameters, but in that case no stock-recruit function is estimated. A warning message is printed if recruit_model > 2 and NAA_re = NULL. If you wish to use a stock-recruit function for expected recruitment, choose recruitment deviations as random effects, either only age-1 (NAA_re = list(sigma="rec")) or all ages (NAA_re = list(sigma="rec+1"), "full state-space" model). See below for details on NAA_re specification.

ecov specifies any environmental covariate data and model. Environmental covariate data need not span the same years as the fisheries data. It can be NULL if no environmental data are to be fit. Otherwise, it must be a named list with the following components:

$label Name(s) of the environmental covariate(s). Used in printing.

$mean Mean observations (matrix). number of years x number of covariates. Missing values = NA.

$logsigma Configure observation standard errors. Options:

Matrix of *log* standard errors with same dimensions as $mean Specified values for each time step

log standard errors for each covariate, numeric vector or matrix w/ dim 1 x n.ecov Specified value the same for all time steps

estimation option (for all covariates). character string: "est_1": Estimated, one value shared among time steps. "est_re": Estimated value for each time step as random effects with two parameters (mean, var)

list of two elements. First is the matrix of log standard errors or the vector of single values for each covariate as above. Second is a character vector of estimation options (NA, "est_1","est_re") for each covariate. For covariates with non-NA values, values in the first element are ignored.

$year Years corresponding to observations (vector of same length as $mean and $logsigma)

$use_obs T/F (or 0/1) vector/matrix of the same dimension as $mean and $logsigma. Use the observation? Can be used to ignore subsets of the ecov without changing data files.

$lag integer vector of offsets between the ecov observation/process and their affect on the stock. I.e. if SST in year *t* affects recruitment in year *t + 1*, set lag = 1. May also be a list (length=n_Ecov) of vectors (length = 2+n_indices) if multiple effects of one or more Ecovs are modeled.

$process_model Process model for the ecov time-series. "rw" = random walk, "ar1" = 1st order autoregressive, NA = do not fit

$where Character vector for where each ecov affects the population. `"recruit"` = recruitment, `"M"` = natural mortality, `"q"` = catchability for indices, `"none"` = fit ecov process model(s) but without an effect on the population. May also be a list (element for each ecov) of character vectors ("none", "recruit", "M", and/or "q") so each ecov can multiple effects.

$indices indices that each ecov affects. Must be a list (length = n_Ecov), where each element is a vector of indices (1:n_indices). Must be provided when any of `where` = "q"

$link_model vector of (orthogonal polynomial order) models for effects of each ecov on the `$where` process. Options: "none", "linear" (default) or "poly-x" where x = 2, ... (e.g. "poly-2" specifies a quadratic model, $b_0 + b_1 * ecov + b_2 * ecov^2 + ...$). Or a list (length = n_Ecov) of character vectors (same options) for modeling effects on (1) recruitment, (2) M, and catchabilities for (3) index 1,..., (2+n_indices) index n_indices (length of the vector is 2 + n_indices).

$ages Ages that each ecov affects. Must be a list of length n.ecov, where each element is a vector of ages. Default = list(1:n.ages) to affect all ages.

$how How does the ecov affect the `$where` process? An integer vector (length = n_Ecov). If corresponding `$where` = "none", then this is ignored. These options are specific to the `$where` process.

Recruitment options (see Iles & Beverton (1998)): = 0 none (but fit ecov time-series model in order to compare AIC)

= 1 "controlling" (dens-indep mortality)

= 2 "limiting" (carrying capacity, e.g. ecov determines amount of suitable habitat)

= 3 "lethal" (threshold, i.e. R –> 0 at some ecov value)

= 4 "masking" (metabolic/growth, decreases dR/dS)

= 5 "directive" (e.g. behavioral)

M options: = 0 none (but fit ecov time-series model in order to compare AIC)

= 1 effect on mean M (shared across ages)

Catchability options: = 0 none (but fit ecov time-series model in order to compare AIC)

= 1 effect on one or more catchabilities (see `$indices`))

`selectivity` specifies options for selectivity, to overwrite existing options specified in the ASAP data file. If `NULL`, selectivity options from the ASAP data file are used. `selectivity` is a list with the following entries:

$model Selectivity model for each block. Vector with length = number of selectivity blocks. Each entry must be one of: "age-specific", "logistic", "double-logistic", or "decreasing-logistic".

$re Time-varying (random effects) for each block. Vector with length = number of selectivity blocks. If `NULL`, selectivity parameters in all blocks are constant over time and uncorrelated. Each entry of `selectivity$re` must be one of the following options, where the selectivity parameters are:

"none" (default) are constant and uncorrelated

"iid" vary by year and age/par, but uncorrelated

"ar1" correlated by age/par (AR1), but not year

"ar1_y" correlated by year (AR1), but not age/par

"2dar1" correlated by year and age/par (2D AR1)

$initial_pars Initial parameter values for each block. List of length = number of selectivity blocks. Each entry must be a vector of length # parameters in the block, i.e. `c(2,0.2)` for logistic or `c(0.5,0.5,0.5,1,1,0.5)` for age-specific with 6 ages. Default is to set at middle of parameter range. This is 0.5 for age-specific and n.ages/2 for logistic, double-logistic, and decreasing-logistic.

$fix_pars Which parameters to fix at initial values. List of length = number of selectivity blocks. E.g. model with 3 age-specific blocks and 6 ages, `list(c(4,5),4,c(2,3,4))` will fix ages 4 and 5 in block 1, age 4 in block 2, and ages 2, 3, and 4 in block 3.

$n_selblocks How many selectivity blocks. Optional. If unspecified and no asap3 object, then this is set to the number of fleets + indices. If specified, ensure other components of `selectivity` are consistent.

`M` specifies estimation options for natural mortality and can overwrite M-at-age values specified in the ASAP data file. If `NULL`, the M-at-age matrix from the ASAP data file is used (M fixed, not estimated). To estimate M-at-age shared/mirrored among some but not all ages, modify `input$map$M_a` after calling `prepare_wham_input` (see vignette for more details). `M` is a list with the following entries:

$model Natural mortality model options are:

- **"constant"** (default) estimate a single mean M shared across all ages
- **"age-specific"** estimate M_a independent for each age
- **"weight-at-age"** specifies M as a function of weight-at-age, $M_y, a = exp(b0 + b1 * log(W_y, a))$, as in Lorenzen (1996) and Miller & Hyun (2018).

$re Time- and age-varying (random effects) on M. Note that random effects can only be estimated if mean M-at-age parameters are (`$est_ages` is not `NULL`).

- **"none"** (default) M constant in time and across ages.
- **"iid"** M varies by year and age, but uncorrelated.
- **"ar1_a"** M correlated by age (AR1), constant in time.
- **"ar1_y"** M correlated by year (AR1), constant all ages.
- **"2dar1"** M correlated by year and age (2D AR1), as in Cadigan (2016).

$initial_means Initial/mean M-at-age vector, with length = number of ages (if `$model = "age-specific"`) or 1 (if `$model = "weight-at-age"` or `"constant"`). If `NULL`, initial mean M-at-age values are taken from the first row of the MAA matrix in the ASAP data file.

$est_ages Vector of ages to estimate M (others will be fixed at initial values). E.g. in a model with 6 ages, `$est_ages = 5:6` will estimate M for the 5th and 6th ages, and fix M for ages 1-4. If `NULL`, M at all ages is fixed at `M$initial_means` (if not `NULL`) or row 1 of the MAA matrix from the ASAP file (if `M$initial_means = NULL`).

$logb_prior (Only if `$model = "weight-at-age"`) TRUE or FALSE (default), should a N(0.305, 0.08) prior be used on log_b? Based on Fig. 1 and Table 1 (marine fish) in Lorenzen (1996).

`NAA_re` specifies options for random effects on numbers-at-age (NAA, i.e. state-space model or not) If `NULL`, a traditional statistical catch-at-age model is fit (NAA = pred_NAA for all ages, deterministic). To fit a state-space model, specify `NAA_re`, a list with the following entries:

$sigma Which ages allow deviations from pred_NAA? Common options are specified with the strings:

"rec" Random effects on recruitment (deviations), all other ages deterministic

"rec+1" "Full state space" model with 2 estimated `sigma_a`, one for recruitment and one shared among other ages

Alternatively, you can specify a more complex structure by entering a vector with length = n.ages, where each entry points to the NAA_sigma to use for that age. E.g. c(1,2,2,3,3,3) will estimate 3 `sigma_a`, with recruitment (age-1) deviations having their own `sigma_R`, ages 2-3 sharing `sigma_2`, and ages 4-6 sharing `sigma_3`.

$cor Correlation structure for the NAA deviations. Options are:

"iid" NAA deviations vary by year and age, but uncorrelated.

"ar1_a" NAA deviations correlated by age (AR1).

"ar1_y" NAA deviations correlated by year (AR1).

"2dar1" NAA deviations correlated by year and age (2D AR1).

$decouple_recruitment T/F determining whether correlation structure of recruitment is independent of RE deviations for older ages (default = FALSE). Only applicable for `NAA_re$sigma = "rec+1"` and correlation across ages is specified. If TRUE and `NAA_re$cor = "ar1_a"`, only deviations for ages>1 have the correlation structure. If TRUE and NAA_re$cor is not "iid" separate year correlation parameters are estimated for recruitment and older ages.

`NAA_re` also can be used to configure initial numbers at age and recruitment models. The optional associated components of `NAA_re` are:

$N1_model Integer determining which way to model the initial numbers at age:

0 (default) age-specific fixed effects parameters

1 2 fixed effects parameters: an initial recruitment and an instantaneous fishing mortality rate to generate an equilibruim abundance at age.

$N1_pars if N1_model = 0, then these would be the initial values to use for abundance at age in the first year. If N1_model = 1, This would be the initial numbers in the first age class and the equilibrium fishing mortality rate generating the rest of the numbers at age in the first year.

$recruit_model Integer determining how to model recruitment. Overrides `recruit_model` argument to `prepare_wham_input`. Must make sure `NAA_re$sigma`, `NAA_re$cor` and `ecov` are properly specified.

1 SCAA, estimating all recruitements as fixed effects or a random walk if NAA_re$sigma specified

2 estimating a mean recruitment with yearly recruitements as random effects

3 Beverton-Holt stock-recruitment with yearly recruitements as random effects

4 Ricker stock-recruitment with yearly recruitements as random effects

$use_steepness T/F determining whether to use a steepness parameterization for a stock-recruit relationship. Only used if recruit_model>2.

$recruit_pars vector of initial parameters for recruitment model. If use_steepness=F, parameters are "alpha" and "beta" otherwise they are steepness and R0.

`catchability` specifies options for catchability. If `NULL` and `asap3` is not NULL, a single catchability parameter for each index is used with initial values specified in ASAP file. If both are NULL, initial catchabilities for all indices = 0.3. Otherwise, it is a list with the following optional entries:

$re Time-varying (random effects) for each index. Vector with length = number of indices. Each entry of `catchability$re` must be one of the following options:

"none" (default) are constant

"iid" vary by year and age/par, but uncorrelated

"ar1" correlated by year (AR1)

$initial_q Initial catchabilities for each index. vector length = number of indices. Will override values provided in `basic_info$q`. If `basic_info$q` and `asap3` are not provided, default q values are 0.3.

$q_lower Lower bound for catchabilities for each index. vector length = number of indices. For indices with NULL components default lower values are 0.

$q_upper Upper bound for catchabilities for each index. vector length = number of indices. For indices with NULL components default lower values are 1000.

$prior_sd vector of NA and standard deviations to use for gaussian prior on logit transform of catchability parameter. Length = number of indices. Indices with non-NA values will have mean logit q as a random effect with mean determined by logit transform of `catchability$initial_q` and sigma as standard error.

`age_comp` specifies the age composition models for fleet(s) and indices. If `NULL`, the multinomial is used because this was the only option in ASAP. The age composition models available are:

`"multinomial"` Multinomial. This is the default because it was the only option in ASAP. 0 parameters.

`"dir-mult"` Saturating Dirichlet-multinomial, parameterized such that effective-sample-size is a nonlinear and saturating function with respect to input-sample-size. 1 parameter. Effective sample size is estimated by the model (Candy 2008)

`"dirichlet-pool0"` Dirichlet, pooling zero observations with adjacent age classes. 1. parameter. See Francis 2014 and Albertsen et al. 2016

`"dirichlet-miss0"` Dirichlet, treating zero observations as missing. 1 parameter.

`"logistic-normal-miss0"` Logistic normal, treating zero observations as missing. 1 parameter.

`"logistic-normal-ar1-miss0"` Logistic normal, treating zero observations as missing. 1 parameter.

`"logistic-normal-pool0"` Logistic normal, pooling zero observations with adjacent age classes. 1 parameter. See Schnute and Haigh (2007) and Francis (2014).

`"logistic-normal-01-infl"` Zero-or-one inflated logistic normal. Inspired by zero-one inflated beta in Ospina and Ferrari (2012). 3 parameters. . No OSA residuals.

`"logistic-normal-01-infl-2par"` Zero-one inflated logistic normal where p0 is a function of binomial sample size. 2 parameters. No OSA residuals.

`"mvtweedie"` Multivariate-tweedie, where the product of composition proportions and input sample sizes follows a distribution with mean equal to the product of predicted proportions and input sample size, and other parameters define the ratio of effective to input sample size (with is bounded 0 to Inf) and the probability of zeros. 2 parameters. No OSA residuals.

`"dir-mult-linear"` Linear Dirichlet-multinomial, parameterized such that effective-sample-size is a linear function with respect to input-sample-size, estimating 1 parameter, $log(\theta)$, where the ratio of effective and input sample size is approximately $\theta/(1+\theta)$, i.e., the logistic transformation of the estimated parameter $log(\theta)$. (Thorson et al. 2017)

The two Dirichlet-multinomial options will only differ when input-sample-size differs among years. In these cases, the linear-Dirichlet multinomial is designed to decrease the effective sample size in each year by approximately the same proportion, while the saturating-Dirichlet multinomial will decrease the years with highest input-sample-size much more than those with lower input-sample-size. One-step-ahead residuals will be calculated for all but the last three options when `do.osa=TRUE` (Nielsen et al. in prep.). An age composition model needs to be specified for each fleet and index. If you would like all fleets and indices to use the same age composition likelihood, you can simply specify one of the strings above, i.e. `age_comp = "logistic-normal-miss0"`. If you do not want the same age composition model to be used for all fleets and indices, you must specify a named list with the following entries:

$fleets A vector of the above strings with length = the number of fleets.

$indices A vector of the above strings with length = the number of indices.

`basic_info` is an optional list of information that can be used to set up the population and types of observations when there is no asap3 file given. Particularly useful for setting up an operating model to simulate population processes and observations. Also can be useful for setting up the structure of assessment model when asap3 has not been used. The current options are:

$ages integer vector of ages (years) with the last being a plus group

$years integer vector of years that the population model spans.

$n_fleets number of fleets.

$agg_catch matrix (length(years) x n_fleets) of annual aggregate catches (biomass) for each fleet.

$catch_paa array (n_fleets x length(years) x n_ages) of each fleet's age composition data (numbers).

$catch_cv matrix (length(years) x n_fleets) of annual CVs for each fleet's aggregate catch observations.

$catch_Neff matrix (length(years) x n_fleets) of annual effective sample sizes for each fleet's age composition observation.

$use_catch_paa 0/1 matrix (length(years) x n_fleets) indicated whether to use each fleet's age composition observation.

$selblock_pointer_fleets integer matrix (length(years) x n_fleets) indicated which selectivity model to use for each fleet each year. Must be consistent with options to `selectivity` option.

$F matrix (length(years) x n_fleets) of annual fishing mortality rates for each fleet to initialize the model.

$n_indices number of indices.

$agg_indices matrix (length(years) x n_indices) of annual aggregate catches (biomass or number) for each fleet.

$index_paa array (n_indices x length(years) x n_ages) of each index's age composition data (biomass or number).

$index_cv matrix (length(years) x n_indices) of annual CVs for each index's aggregate observations.

$index_Neff matrix (length(years) x n_indices) of annual effective sample sizes for each index's age composition observation.

$units_indices 1/2 matrix (length = n_indices) indicated whether indices are in biomass or numbers, respectively.

$units_index_paa 1/2 matrix (length = n_indices) indicated whether to use each index's age composition observation are in numbers or biomass.

$use_index_paa 0/1 matrix (length(years) x n_indices) indicated whether to use each index's age composition observation.

$selblock_pointer_indices integer matrix (length(years) x n_indices) indicated which selectivity model to use for each index each year. Must be consistent with options to `selectivity` option.

$fracyr_indices matrix (length(years) x n_indices) of annual proportions of the year elapsed when each index is observing the population.

$waa array ((n_fleets + n_indices + 2) x length(years) x length(ages)) of annual weight at at age for each fleet, each index, total catch, and spawning biomass.

$maturity matrix (length(years) x length(ages)) of annual maturity at age for estimating spawning biomass.

$fracyr_SSB vector (1 or length(years)) of yearly proportions (0-1) of the year at which to calculate spawning biomass.

$Fbar_ages integer vector of ages to use to average total F at age defining fully selected F for reference points. May not be clearly known until a model is fitted.

$q vector (length(n_indices)) of catchabilities for each of the indices to initialize the model.

$percentSPR (0-100) percentage of unfished spawning biomass per recruit for determining equilibrium fishing mortality reference point

$percentFXSPR (0-100) percentage of SPR-based F to use in projections.

$percentFMSY (0-100) percentage of Fmsy to use in projections.

$XSPR_input_average_years which years to average inputs to per recruit calculation (selectivity, M, WAA, maturity) for SPR-based reference points. Default is last 5 years (tail(1:length(years),5))

$XSPR_R_avg_yrs which years to average recruitments for calculating SPR-based SSB reference points. Default is 1:length(years)

$XSPR_R_opt 1(3): use annual R estimates(predictions) for annual SSB_XSPR, 2(4): use average R estimates(predictions). 5: use bias-corrected expected recruitment

$simulate_process_error T/F vector (length = 5). When simulating from the model, whether to simulate any process errors for NAA, M, selectivity, Ecov, and q. Only used if applicable.

$simulate_observation_error T/F vector (length = 3). When simulating from the model, whether to simulate catch, index, and ecov observations.

$simulate_period T/F vector (length = 2). When simulating from the model, whether to simulate base period (model years) and projection period.

$bias_correct_process T/F. Perform bias correction of log-normal random effects for NAA.

$bias_correct_observation T/F. Perform bias correction of log-normal observations.

$bias_correct_BRPs T/F. Perform bias correction of analytic SSB/R and Y/R when there is bias correction of log-normal NAA.

If other arguments to `prepare_wham_input` are provided such as `selectivity`, `M`, and `age_comp`, the information provided there must be consistent with `basic_info`. For example the dimensions for number of years, ages, fleets, and indices.

Value

a named list with the following components:

`data` Named list of data, passed to `TMB::MakeADFun`

`par` Named list of parameters, passed to `TMB::MakeADFun`

`map` Named list defining how to optionally collect and fix parameters, passed to `TMB::MakeADFun`

`random` Character vector of parameters to treat as random effects, passed to `TMB::MakeADFun`

`years` Numeric vector of years to fit WHAM model (specified in ASAP3 .dat file)

`ages.lab` Character vector of age labels, ending with plus-group (specified in ASAP3 .dat file)

`model_name` Character, name of stock/model (specified in call to `prepare_wham_input`)

See Also

`read_asap3_dat`, `fit_wham`, ASAP, Iles & Beverton (1998)

Examples

```
## Not run:
asap3 = read_asap3_dat("ex1_SNEMAYT.dat")
input = prepare_wham_input(asap3)
mod = fit_wham(input)

# no ASAP3 file, default parameter values and configuration
input = prepare_wham_input()
mod = fit_wham(input, fit = FALSE)
set.seed(8675309)
simdata = mod$simulate(complete=TRUE)
input$data = simdata
fit = fit_wham(input, do.osa = FALSE)

## End(Not run)
```

`project_wham` *Project a fit WHAM model*

Description

Provides projections/forecasts for an existing (already fit) WHAM model.

Usage

```
project_wham(
  model,
  proj.opts = list(n.yrs = 3, use.last.F = TRUE, use.avg.F = FALSE, use.FXSPR = FALSE,
    use.FMSY = FALSE, proj.F = NULL, proj.catch = NULL, avg.yrs = NULL, cont.ecov = TRUE,
    use.last.ecov = FALSE, avg.ecov.yrs = NULL, proj.ecov = NULL, cont.Mre = NULL,
    avg.rec.yrs = NULL, percentFXSPR = 100, percentFMSY = 100, proj_F_opt = NULL,
     proj_Fcatch = NULL),
  n.newton = 3,
  do.sdrep = TRUE,
  MakeADFun.silent = FALSE,
  save.sdrep = TRUE
)
```

Arguments

model a previously fit wham model

proj.opts a named list with the following components:

- `$n.yrs` (integer), number of years to project/forecast. Default = `3`.
- `$use.last.F` (T/F), use terminal year F for projections. Default = `TRUE`.
- `$use.avg.F` (T/F), use average of F over certain years for projections. Default = `FALSE`. Years to average over determined by $avg.yrs defined below.
- `$use.FXSPR` (T/F), calculate and use F at X% SPR for projections. Default = `FALSE`.
- `$use.FMSY` (T/F), calculate and use FMSY for projections. Default = `FALSE`.
- `$proj.F` (vector), user-specified fishing mortality for projections. Length must equal `n.yrs`.
- `$proj.catch` (vector), user-specified aggregate catch for projections. Length must equal `n.yrs`.
- `$avg.yrs` (vector), specify which years to average over for calculating reference points. Default = last 5 model years, `tail(model$years, 5)`.
- `$cont.ecov` (T/F), continue ecov process (e.g. random walk or AR1) for projections. Default = `TRUE`.
- `$use.last.ecov` (T/F), use terminal year ecov for projections.
- `$avg.ecov.yrs` (vector), specify which years to average over the environmental covariate(s) for projections.
- `$proj.ecov` (matrix), user-specified environmental covariate(s) for projections. `n.yrs` x `n.ecov`.
- `$cont.Mre` (T/F), continue M random effects (i.e. AR1_y or 2D AR1) for projections. Default = `TRUE`. If `FALSE`, M will be averaged over `$avg.yrs` (which defaults to last 5 model years).
- `$avg.rec.yrs` (vector), specify which years to calculate the CDF of recruitment for use in projections. Default = all model years. Only used when recruitment is estimated as fixed effects (SCAA).
- `$percentFXSPR` (scalar), percent of F_XSPR to use for calculating catch in projections, only used if $use.FXSPR = TRUE. For example, GOM cod uses F = 75% F_40%SPR, so `proj.opts$percentFXSPR = 75`. Default = 100.
- `$percentFMSY` (scalar), percent of F_MSY to use for calculating catch in projections, only used if $use.FMSY = TRUE.
- `$proj_F_opt` (vector), integers specifying how to configure each year of the projection: 1: use terminal F, 2: use average F, 3: use F at X% SPR, 4:

use specified F, 5: use specified catch, 6: use Fmsy. Overrides any of the above specifications.

- `$proj_Fcatch` (vector), catch or F values to use each projection year: values are not used when using Fmsy, FXSPR, terminal F or average F. Overrides any of the above specifications of proj.F or proj.catch.
- `$proj_mature` (matrix), user-supplied maturity values for the projection years with dimensions (`n.yrs` x n_ages).
- `$proj_waa` (3-d array), user-supplied waa values for the projection years with first and third dimensions equal to that of `model$input$data$waa` (waa source x `n.yrs` x n_ages).
- `$proj_R_opt` (integer), 1: continue any RE processes for recruitment, 2: make projected recruitment consistent with average recruitment in SPR reference points and cancel any bias correction for NAA in projection years.

`n.newton` integer, number of additional Newton steps after optimization. Passed to `fit_tmb`. Default = `0` for projections.

`do.sdrep` T/F, calculate standard deviations of model parameters? See `sdreport`. Default = `TRUE`.

`MakeADFun.silent`

T/F, Passed to silent argument of `TMB::MakeADFun`. Default = `FALSE`.

`save.sdrep` T/F, save the full `TMB::sdreport` object? If `FALSE`, only save `summary.sdreport` to reduce model object file size. Default = `TRUE`.

Details

WHAM implements five options for handling fishing mortality in the projections. Exactly one of these must be specified in `proj.opts`:

- Use last year F (default). Set `proj.opts$use.last.F = TRUE`. WHAM will use F in the terminal model year for projections.
- Use average F. Set `proj.opts$use.avg.F = TRUE`. WHAM will use F averaged over `proj.opts$avg.yrs` for projections (as is done for M-, maturity-, and weight-at-age).
- Use F at X% SPR. Set `proj.opts$use.FXSPR = TRUE`. WHAM will calculate F at X% SPR.
- Specify F. Provide `proj.opts$proj.F`, an F vector with length = `n.yrs`.
- Specify catch. Provide `proj.opts$proj.catch`, a vector of aggregate catch with length = `n.yrs`. WHAM will calculate F to get specified catch.

`proj.opts$avg.yrs` controls which years the following will be averaged over in the projections:

- Maturity-at-age
- Weight-at-age
- Natural mortality-at-age
- Fishing mortality-at-age (if `proj.opts$use.avgF = TRUE`)

If fitting a model with recruitment estimated freely in each year, i.e. as fixed effects as in ASAP, WHAM handles recruitment in the projection years similarly to using the empirical cumulative distribution function. WHAM does this by calculating the mean and standard deviation of log(R) over all model years (default) or a specified subset of years (`proj.opts$avg.rec.yrs`). WHAM then treats recruitment in the projections as a random effect with this mean and SD, i.e. log(R) ~ N(meanlogR, sdlogR).

WHAM implements four options for handling the environmental covariate(s) in the projections. Exactly one of these must be specified in `proj.opts` if `ecov` is in the model:

(Default) Continue ecov process model (e.g. random walk, AR1) Set `$cont.ecov = TRUE`. WHAM will estimate the ecov process in projection years (i.e. continue the random walk / AR1 process).

Use last year ecov(s) Set `$use.last.ecov = TRUE`. WHAM will use ecov value from the terminal year (of population model) for projections.

Use average ecov(s) Provide `$avg.yrs.ecov`, a vector specifying which years to average over the environmental covariate(s) for projections.

Specify `ecov` Provide `$proj.ecov`, a matrix of user-specified environmental covariate(s) to use for projections. Dimensions must be # projection years (`proj.opts$n.yrs`) x # ecovs (`ncols(ecov$mean)`).

If the original model fit the ecov in years beyond the population model, WHAM will use the already-fit ecov values for the projections. If the ecov model extended at least `proj.opts$n.yrs` years beyond the population model, then none of the above need be specified.

Value

a projected WHAM model with additional output if specified:

`$rep` List of derived quantity estimates (see examples)

`$sdrep` Parameter estimates (and standard errors if `do.sdrep=TRUE`)

`$peels` Retrospective analysis (if `do.retro=TRUE`)

`$osa` One-step-ahead residuals (if `do.osa=TRUE`)

See Also

`fit_wham`, `fit_tmb`

Examples

```
## Not run:
data("input4_SNEMAYT") # load SNEMA yellowtail flounder input data and model settings
mod <- fit_wham(input4_SNEMAYT) # using default values (do.proj=T)

mod2 <- fit_wham(input4_SNEMAYT, do.retro=F, do.osa=F, do.proj=F) # fit model without projections, retro analysis,
mod_proj <- project_wham(mod2) # add projections to previously fit model, using default values: use.lastF = TRUE, n.

names(mod_proj$rep) # list of derived quantities
tail(mod_proj$rep$SSB, 3) # get 3-year projected SSB estimates (weight, not numbers)

x = summary(mod_proj$sdrep)
unique(rownames(x)) # list of estimated parameters and derived quanitites with SE
x = x[rownames(x) == "log_SSB",] # SSB estimates with SE
ssb.mat = exp(cbind(x, x[,1] + qnorm(0.975)*cbind(-x[,2],x[,2])))/1000 # calculate 95% CI
```

```
colnames(ssb.mat) <- c("SSB","SSB_se","SSB_lower","SSB_upper")
tail(ssb.mat, 3) # 3-year projected SSB estimates with SE and 95% CI

## End(Not run)
```

read_asap3_dat *Read an ASAP3 .dat file into R*

Description

WHAM is built on ASAP ([Legault and Restrepo (1999)](http://sedarweb.org/docs/wsupp/S12RD06%20ASAPdoc.pdf)) and this function provides functionality to use a preexisting ASAP3 input data file. The output of `read_asap3_dat` should then be passed to `prepare_wham_input`. If you are not familiar with ASAP3 input files, see the ASAP documentation and code.

Usage

```
read_asap3_dat(filename)
```

Arguments

`filename`	character, name of ASAP3 .dat file. The file either needs to be in the current working directory, or `filename` can include the path.

Value

a named list with the following components:

`dat` Named list of input data and parameters

`comments` Comments at top of ASAP3 .dat file (indicated by "#")

See Also

`prepare_wham_input`, `fit_wham`, ASAP documentation

Examples

```
## Not run:
asap3 = read_asap3_dat("ASAP_SNEMAYT.dat")
input = prepare_wham_input(asap3)
mod = fit_wham(input)

## End(Not run)
```

read_asap3_fit *Read ASAP3 fit*

Description

Gets output from a fit ASAP3 model for plotting with WHAM models.

Usage

```
read_asap3_fit(wd, asap.name, pSPR = 40)
```

Arguments

`wd`	character, directory where ASAP3 output files are located (ex: 'C:/MY/file/directories/model/'). 5 files are needed: `.rdat`, `.dat`, `.std`, `.cor`, and `.par`.
`asap.name`	character, base name of original .dat file (i.e. without the .dat extension)
`pSPR`	scalar, user-specified percent SPR to use for reference points, expressed as 100*SSBPR(Fspr)/SSBPR(F=0). Default = 40.

Value

a named list with the following elements:

`$years` numeric vector, model years only, e.g. `1972:2020`

`$years_full` numeric vector, model + proj years, e.g. `1972:2022`. For ASAP this will be the same as `$years`.

`$selAA` list of length(n_selblocks), first the fleet blocks then indices, i.e. if 4 fleet blocks and 3 indices, `selAA[[5]]` is for index 1. Each element is a matrix, years (rows) x ages (cols), selectivity at age

`$selblock_pointer_fleets` matrix, n_years x n_fleets, indices of selAA used by each fleet in each year

`$selblock_pointer_indices` matrix, n_years x n_indices, indices of selAA used by each index in each year

`$MAA` matrix, n_years x n_ages, natural mortality

`$log_SSB` matrix, n_years x 2, log-scale spawning stock biomass. 1st col = MLE, 2nd col = SE (from .std file in ADMB).

`$log_F` matrix, n_years x 2, log-scale fully-selected F. 1st col = MLE, 2nd col = SE (from .std file in ADMB).

`$log_NAA` matrix, n_years x n_ages, numbers at age

`$NAA_CV` matrix, n_years x n_ages, CV of numbers at age

`$percentSPR` scalar, X% SPR used to calculate reference points, default = 40

`$log_Y_FXSPR` matrix, n_years x 2, log-scale yield at FXSPR. 1st col = MLE, 2nd col = SE.

`$log_FXSPR` matrix, n_years x 2, log-scale FXSPR. 1st col = MLE, 2nd col = SE.

$log_SSB_FXSPR matrix, n_years x 2, log-scale SSB at FXSPR, i.e. annual numerator of SPR(FXSPR) * Recruits. 1st col = MLE, 2nd col = SE.

$log_rel_ssb_F_cov list, length n_years, each element is a 2x2 covariance matrix with SSB / SSB_FXSPR first and F / F_FXSPR second

See Also

compare_wham_models, read_wham_fit

Examples

```
## Not run:
base <- read_asap3_fit(wd=file.path(getwd(),'asap_results'), asap.name='BASE_5C.DAT', pSPR=40)
m1 <- fit_wham(input1)
m2 <- fit_wham(input2)
mods <- list(base=base, m1=m1, m2=m2)
res <- compare_wham_models(mods)

## End(Not run)
```

read_wham_fit *Read WHAM fit*

Description

Gets output from a fit WHAM model for plotting with other models. Internal function, called within compare_wham_models.

Usage

```
read_wham_fit(mod, alphaCI = 0.05)
```

Arguments

mod	output from fit_wham
alphaCI	(1-alpha)% confidence intervals will be calculated. Default = 0.05 for 95% CI.

Value

a named list with the following elements:

$years numeric vector, model years only, e.g. 1972:2020

$years_full numeric vector, model + proj years, e.g. 1972:2022

$selAA list of length(n_selblocks), first the fleet blocks then indices, i.e. if 4 fleet blocks and 3 indices, selAA[[5]] is for index 1. Each element is a matrix, years (rows) x ages (cols), selectivity at age

$selblock_pointer_fleets matrix, years x fleets, indices of selAA used by each fleet in each year

$selblock_pointer_indices matrix, years x indices, indices of selAA used by each index in each year

$MAA matrix, years x ages, natural mortality

$log_SSB matrix, years x 2, log-scale spawning stock biomass. 1st col = MLE, 2nd col = SE.

$log_F matrix, years x 2, log-scale fully-selected F. 1st col = MLE, 2nd col = SE.

$log_NAA matrix, years x ages, numbers at age

$NAA_CV matrix, years x ages, CV of numbers at age

$percentSPR scalar, X% SPR used to calculate reference points, default = 40

$log_Y_FXSPR matrix, years x 2, log-scale yield at FXSPR. 1st col = MLE, 2nd col = SE.

$log_FXSPR matrix, years x 2, log-scale FXSPR. 1st col = MLE, 2nd col = SE.

$log_SSB_FXSPR matrix, years x 2, log-scale SSB at FXSPR. 1st col = MLE, 2nd col = SE.

$log_rel_ssb_F_cov list, length n_years, each element is a 2x2 covariance matrix with SSB/SSB_FXSPR first and F/F_FXSPR second

See Also

fit_wham, read_asap3_fit, compare_wham_models

---

retro *Run retrospective analysis*

Description

Internal function called by fit_wham. Calls fit_peel to fit the model peeling off 1, 2, ..., n.peels years of data.

Usage

```
retro(
  model,
  n.peels = 7,
  ran = "log_NAA",
  do.sdrep = FALSE,
  n.newton = 0,
  MakeADFun.silent = FALSE,
  retro.silent = FALSE,
  save.input = FALSE
)
```

Arguments

`model` Optimized TMB model, output from `fit_tmb`.

`n.peels` Integer, number of peels to use in retrospective analysis. Default = `7`.

`ran` Character, specifies which parameters to treat as random effects. Default = `"log_NAA"`.

`do.sdrep` T/F, calculate standard deviations of model parameters for each peel? Default = `FALSE`.

`n.newton` integer, number of additional Newton steps after optimization for each peel. Default = `0`.

`MakeADFun.silent`
T/F, Passed to silent argument of `TMB::MakeADFun`. Default = `FALSE`.

`retro.silent` T/F, Passed to argument of internal fit_peel function. Determines whether peel number is printed to screen. Default = `FALSE`.

`save.input` T/F, should modified input list be saved for every peel? Necessary to project from a peel but increases model object size. Default = `FALSE`.

Value

`peels`, a list of length `n.peels`, where entry *i* is a model fit by peeling off *i* years of data.

See Also

`fit_wham`, `fit_peel`

`retro_res` *Extract retrospective results for plotting*

Description

Extract retrospective results for plotting

Usage

```
retro_res(model)
```

Arguments

`model` A fit WHAM model, output from `fit_wham` with `do.retro` = `TRUE`.

Value

a named list with the components:

`SSB` Spawning stock biomass

`Fbar` Fishing mortality

`NAA` Numbers-at-age

See Also

fit_wham, retro

Examples

```
## Not run:
data("input4_SNEMAYT") # load SNEMA yellowtail flounder data and parameter settings
mod = fit_wham(input4_SNEMAYT) # using default values: do.retro = T, n.peels = 7
x = retro_res(mod) # get retrospective results

## End(Not run)
```

set_age_sel0 *Make one or more selectivity blocks with age-specific parameters*

Description

Make one or more selectivity blocks with age-specific parameters

Usage

```
set_age_sel0(input, selblocks)
```

Arguments

input	list containing data and parameters (output from prepare_wham_input)
selblocks	numeric, number of age-specific selectivity blocks

Value

a modified list of data and parameters

Examples

```
## Not run:
asap3 = read_asap3_dat("ASAP_SNEMAYT.dat")
input = prepare_wham_input(asap3)
input = set_age_sel0(input, selblocks = 1:3)
mod = fit_wham(input)

## End(Not run)
```

wham_html — *Create HTML file to view output plots in browser*

Description

Writes a set of HTML files with tabbed navigation between them. Called by `plot_wham_output` if ‘out.type = ’html’‘ (default). Opens main file in default browser. Modified from [‘r4ss::SS_html‘](https://github.com/r4ss/r4ss

Usage

```
wham_html(dir.main = NULL, title = "WHAM Output", width = 500, openfile = TRUE)
```

Arguments

`dir.main`	directory to save html file (`plot_wham_output` makes ‘.png‘ plot files in a ‘plots_png‘ subdirectory of ‘dir.main‘).
`title`	Title for HTML page.
`width`	Width of plots (in pixels).
`openfile`	Automatically open index.html in default browser?

See Also

`plot_wham_output`, ‘r4ss::SS_html()‘

%>% — *Pipe function*

Description

Allows use of the pipe function, `%>%`

Index