

Package: wham (via r-universe)

October 28, 2026

Title Woods Hole Assessment Model (WHAM)

Version 2.1.0.9012

Description The Woods Hole Assessment Model (WHAM) is a general age-structured stock assessment framework that can be configured to estimate assessment models that range in complexity from statistical catch-at-age (SCAA) model with annual recruitments as fixed effects, to state-space, multi-stock, multi-region, age-structured models where many parameters can be treated as time- and age-varying process errors and/or allowing effects of environmental covariates. WHAM is a generalization of 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, although this is not a requirement. Many of the plotting functions for input data, results, and diagnostics have modified from code written by Chris Legault and Liz Brooks.

Depends R (>= 4.2.2)

Imports Deriv (>= 4.1.6), dplyr (>= 1.1.4), ellipse (>= 0.5.0), ggplot2 (>= 3.5.1), kableExtra (>= 1.4.0), knitr (>= 1.48), mnormt (>= 2.1.1), rmarkdown (>= 2.28), tidyr (>= 1.3.1), TMB (>= 1.9.17), viridis (>= 0.6.5), viridisLite (>= 0.4.2)

Remotes kaskr/adcomp/TMB

LinkingTo TMB, RcppEigen

URL <https://timjmiller.github.io/wham/>,
<https://github.com/timjmiller/wham>,
<https://doi.org/10.1016/j.fishres.2021.105967>,
<https://doi.org/10.1139/cjfas-2025-0097>

BugReports <https://github.com/timjmiller/wham/issues>

License GPL-3

LazyData true

VignetteBuilder knitr

Encoding UTF-8

Suggests testthat, devtools, here, parallel, snowfall

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libpng-dev libuv1-dev libxml2-dev

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

Date/Publication 2026-10-28 15:57:12 UTC

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

RemoteRef HEAD

RemoteSha ffcbfd093161e8210aa20716933975191b661807

Contents

aic	3
check_convergence	4
check_estimability	5
compare_wham_models	5
do_reference_points	8
do_retro_peels	8
do_sdreport	9
extract_fixed	10
fit_peel	11
fit_tmb	12
fit_wham	13
jitter_wham	16
make_osa_residuals	17
mohns_rho	18
plot_wham_output	19
prepare_projection	20
prepare_wham_input	24
project_wham	28
read_asap3_dat	33
read_asap3_fit	34
read_wham_fit	36
reduce_input	37
retro	37
self_test	38
set_age_comp	40
set_catch	42
set_ecov	43
set_F	46
set_indices	47
set_L	48
set_M	49

aic

3

set_move

51

set_NAA

53

set_osa_obs

56

set_q

56

set_selectivity

58

wham_html

61

Index

62

aic	<i>Calculate Akaike's Information Criterion for a WHAM model</i>
-----	--

Description

Calculates AIC from a fitted WHAM model's negative log-likelihood and number of estimated parameters.

Usage

```
aic(mod, conditional = FALSE)
```

Arguments

- mod

A fitted WHAM model object returned by `fit_wham()`.
- conditional

(TRUE/FALSE) When the model includes random effects, the default (`conditional = FALSE`) calculation uses the marginal likelihood and number of fixed effects parameters. If `conditional = TRUE`, the joint likelihood of the data conditional on the estimated random effects is used with an estimated effective degrees of freedom that is calculated using the approach described by [Zhang et al. 2024](#) and code provided by Noel Cadigan.

Value

A numeric AIC value with attributes denoting the degrees of freedom, number of observations, and the type (marginal or conditional).

Examples

```
## Not run:
mod <- fit_wham(input)
aic(mod)
aic(mod, conditional = TRUE)

## End(Not run)
```

check_convergence	<i>Check convergence of WHAM model</i>
-------------------	--

Description

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

Usage

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

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	<i>Check for identifiability of fixed effects Originally provided by the R</i> <i>https://github.com/kaskr/TMB_contrib_R/tree/master/TMBhelper</i> <i>package. Internal function called by <code>fit_tmb</code>.</i>
--------------------	---

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-invertible 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	<i>Compare multiple WHAM (or ASAP) models</i>
---------------------	---

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(),
  compare.opts = NULL,
  table.opts = NULL,
  plot.opts = NULL,
  fname = NULL,
  sort = NULL,
```

```

    calc.rho = NULL,
    calc.aic = NULL,
    do.print = NULL
  )

```

Arguments

<code>mods</code>	(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.
<code>do.table</code>	T/F, produce table of AIC and/or Mohn's rho? Default = TRUE.
<code>do.plot</code>	T/F, produce plots? Default = TRUE.
<code>fdir</code>	character, path to directory to save table and/or plots. Default = <code>getwd()</code> .
<code>compare.opts</code>	list of options to generate comparison results: <code>\$stock</code> integer, which stock to include in results. Default = 1. <code>\$region</code> integer, which region to include in results. Default = 1.
<code>table.opts</code>	list of options for AIC/rho table: <code>\$fname</code> character, filename to save CSV results table (.csv will be appended). Default = 'model_comparison'. <code>\$sort</code> T/F, sort by AIC? Default = TRUE. <code>\$calc.rho</code> T/F, calculate Mohn's rho? Retrospective analysis must have been run for all modes. Default = TRUE. <code>\$calc.aic</code> T/F, calculate AIC? Default = TRUE. <code>\$print</code> T/F, print table to console? Default = TRUE. <code>\$save.csv</code> T/F, save table as a CSV file? Default = FALSE.
<code>plot.opts</code>	list of options for plots: <code>\$out.type</code> character, either 'pdf' or 'png' (default = 'png' because I am not sure <code>system('pdftk')</code> will work across platforms.) <code>\$ci</code> vector of T/F, length = 1 (applied to all models) or number of models <code>\$years</code> vector, which years to plot? Default = all (model and projection years). <code>\$which</code> vector, which plots to make? Default = all. See details. <code>\$relative.to</code> character, name of "base" model to plot differences relative to. <code>\$alpha</code> scalar, (1-alpha)% confidence intervals will be plotted. Default = 0.05 for 95% CI. <code>\$ages.lab</code> vector, overwrite model age labels. <code>\$kobe.yr</code> integer vector (length = 1 (applied to all models) or number of models, which year to use in Kobe plot (relative status). Default = terminal model year(s). <code>\$M.age</code> integer, which age to use in M time-series plot. Default = <code>max(data\$which_F_age)</code> (max age of F to use for full total F). <code>\$return.ggplot</code> T/F, return a list of ggplot2 objects for later modification? Default = TRUE. <code>\$kobe.prob</code> T/F, print probabilities for each model in each quadrant of Kobe plot? Default = TRUE. <code>\$refpt</code> "XSPR" or "MSY", which reference point to use. Default = "XSPR". <code>\$browse</code> Open html document in web browser (if <code>\$out.type</code> = 'png') (default = TRUE).

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)
```

do_reference_points	<i>Add reporting of biological reference points to WHAM model</i>
---------------------	---

Description

Changes internal flags to do the extra calculations and reporting for reference points.

Usage

```
do_reference_points(model, do.sdrep = FALSE, save.sdrep = TRUE)
```

Arguments

model	a fitted WHAM model object returned by <code>fit_wham</code> or <code>project_wham</code> .
do.sdrep	T/F, calculate standard deviations of model parameters? See sdreport . Default = FALSE.
save.sdrep	T/F, save the full <code>TMB:sdreport</code> object? If FALSE, only save <code>summary.sdreport</code> to reduce model object file size. Default = TRUE.

See Also

[fit_wham](#), [project_wham](#)

do_retro_peels	<i>Fit retrospective peels and add them to the fitted model object</i>
----------------	--

Description

Function to add fitted retrospective peels to fitted model returned by `fit_wham`. Calls `retro`. to fit the model peeling off 1, 2, ..., n.peels years of data. This is just a wrapper for `retro` that instead of returning just the list of peels, returns the fitted model with the peels.

Usage

```
do_retro_peels(
  model,
  n.peels = 7,
  ran = NULL,
  use.mle = TRUE,
  do.sdrep = FALSE,
  n.newton = 0,
  MakeADFun.silent = FALSE,
  retro.silent = FALSE,
  save.input = FALSE,
  do.brps = FALSE,
```

```

    check.version = TRUE,
    save.sdrep = FALSE
  )

```

Arguments

<code>model</code>	Optimized TMB model, output from <code>fit_tmb</code> .
<code>n.peels</code>	Integer, number of peels to use in retrospective analysis. Default = 7.
<code>ran</code>	Character, specifies which parameters to treat as random effects. Default = <code>"model\$input\$random"</code> .
<code>use.mle</code>	T/F, use MLEs from full model fit as initial values for each peel? If not, the initial values from full model input are used. Default = TRUE.
<code>do.sdrep</code>	T/F, calculate standard deviations of model parameters for each peel? Default = FALSE.
<code>n.newton</code>	integer, number of additional Newton steps after optimization for each peel. Default = 0.
<code>MakeADFun.silent</code>	T/F, Passed to silent argument of <code>TMB::MakeADFun</code> . Default = FALSE.
<code>retro.silent</code>	T/F, Passed to argument of internal <code>fit_peel</code> function. Determines whether peel number is printed to screen. Default = FALSE.
<code>save.input</code>	T/F, should modified input list be saved for every peel? Necessary to project from a peel but increases model object size. Default = FALSE.
<code>do.brps</code>	T/F, calculate and report biological reference points
<code>check.version</code>	T/F, whether to verify the wham package commit and version for the fitted model are the same as the currently used package.
<code>save.sdrep</code>	T/F, save the full <code>TMB::sdreport</code> object? If FALSE, only save <code>summary.sdreport</code> to reduce model object file 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`, `retro`, `fit_peel`

`do_sdreport`

Add TMB sdreport object to WHAM model

Description

Runs `TMB::sdreport` and adds the object to the fitted (and projected) model list. E.g., `fit$sdrep`.

Usage

```
do_sdreport(
  model,
  save.sdrep = TRUE,
  TMB.bias.correct = FALSE,
  TMB.jointPrecision = FALSE
)
```

Arguments

`model` a fitted WHAM model object returned by `fit_wham` or `project_wham`.

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

`TMB.bias.correct` T/F whether to use the `bias.correct` feature of `TMB::sdreport`. Default = FALSE.

`TMB.jointPrecision` T/F whether to return the joint precision matrix for the fixed and random effects from `TMB::sdreport`. Default = FALSE.

See Also

[fit_wham](#), [project_wham](#)

<code>extract_fixed</code>	<i>Extract fixed effects Originally provided by the Rhrefhttps://github.com/kaskr/TMB_contrib_R/tree/master/TMBhelper package. Internal function called by check_estimability.</i>
----------------------------	---

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

<code>peel</code>	Integer, number of years of data to remove before model fitting.
<code>input</code>	input with same structure as that provided by <code>prepare_wham_input</code> . May want to use <code>input\$par = model\$parList</code> to start at MLEs.
<code>do.sdrep</code>	T/F, calculate standard deviations of model parameters? Default = FALSE.
<code>n.newton</code>	integer, number of additional Newton steps after optimization for each peel. Default = 3.
<code>MakeADFun.silent</code>	T/F, Passed to silent argument of <code>TMB::MakeADFun</code> . Default = FALSE.
<code>retro.silent</code>	T/F, Passed to argument of internal <code>fit_peel</code> function. Determines whether peel number is printed to screen. Default = FALSE.
<code>save.input</code>	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,
  use.optim = FALSE,
  opt.control = NULL
)
```

Arguments

<code>model</code>	Output from <code>TMB::MakeADFun</code> .
<code>n.newton</code>	Integer, number of additional Newton steps after optimization. Default = 3.
<code>do.sdrep</code>	T/F, calculate standard deviations of model parameters? See <code>TMB::sdreport</code> . Default = TRUE.
<code>do.check</code>	T/F, check if model parameters are identifiable? Runs internal <code>check_estimability</code> , originally provided by the <code>TMBhelper</code> package. Default = TRUE.
<code>save.sdrep</code>	T/F, save the full <code>TMB::sdreport</code> object? If FALSE, only save <code>summary.sdreport</code> to reduce model object file size. Default = FALSE.
<code>use.optim</code>	T/F, use <code>stats::optim</code> instead of <code>stats::nlminb</code> ? Default = FALSE.
<code>opt.control</code>	list of control parameters to pass to optimizer. For <code>nlminb</code> default = <code>list(iter.max = 1000, eval.max = 1000)</code> . For <code>optim</code> default = <code>list(maxit=1000)</code> .

Value

`model`, appends the following:

```
model$opt  Output from stats::nlminb
model$date System date
model$dir  Current working directory
model$rep  model$report(model$env$last.par.best)
model$TMB_version  Version of TMB installed
model$parList  List of parameters, model$env$parList(x = model$opt$par, par = model$env$last.par.best)
model$final_gradient  Final gradient, model$gr(model$opt$par)
model$sdrep  Estimated standard deviations for model parameters, TMB::sdreport or summary.sdreport)
```

See Also

[fit_wham](#), [retro](#), [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 = "oneStepGaussianOffMode", 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,
  do.brps = TRUE,
  fit.tmb.control = NULL,
  TMB.bias.correct = FALSE,
  TMB.jointPrecision = FALSE
)
```

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 .
-------	--

	<code>\$random</code> Character vector defining the parameters to treat as random effect. See MakeADFun .
	<code>\$years</code> Numeric vector of the years which the model spans. Not important for model fitting, but useful for plotting.
	<code>\$model_name</code> Character, name of the model, e.g. "Yellowtail flounder"
	<code>\$ages.lab</code> Character vector of the age labels, e.g. <code>c("1", "2", "3", "4+")</code> .
<code>n.newton</code>	integer, number of additional Newton steps after optimization. Passed to fit_tmb . Default = 3.
<code>do.sdrep</code>	T/F, calculate standard deviations of model parameters? See sdreport . Default = TRUE.
<code>do.retro</code>	T/F, do retrospective analysis? Default = TRUE.
<code>n.peels</code>	integer, number of peels to use in retrospective analysis. Default = 7.
<code>do.osa</code>	T/F, calculate one-step-ahead (OSA) residuals? Default = TRUE. See details. Returned as <code>mod\$osa\$residual</code> .
<code>osa.opts</code>	list of 2 options (method, parallel) for calculating OSA residuals, passed to TMB::oneStepPredict . Default: <code>osa.opts = list(method="oneStepGaussianOffMode", parallel=TRUE)</code> . See make_osa_residuals .
<code>do.post.samp</code>	T/F, obtain sample from posterior of random effects? Default = TRUE. NOT YET IMPLEMENTED.
<code>model</code>	(optional), a previously fit wham model.
<code>do.check</code>	T/F, check if model parameters are identifiable? Passed to fit_tmb . Runs internal function <code>check_estimability</code> , originally provided by https://github.com/kaskr/TMB_contrib_R/TM . Default = TRUE.
<code>MakeADFun.silent</code>	T/F, Passed to silent argument of TMB::MakeADFun . Default = FALSE.
<code>retro.silent</code>	T/F, Passed to argument of internal retro function. Determines whether peel number is printed to screen. Default = FALSE.
<code>do.proj</code>	T/F, do projections? Default = FALSE. If true, runs project_wham .
<code>proj.opts</code>	a named list with the following components: • <code>\$n.yrs</code> (integer), number of years to project/forecast. Default = 3. • <code>\$use.last.F</code> (T/F), use terminal year F for projections. Default = TRUE. • <code>\$use.FXSPR</code> (T/F), calculate and use F at X • <code>\$use.FMSY</code> (T/F), calculate and use FMSY for projections. • <code>\$proj.F</code> (vector), user-specified fishing mortality for projections. Length must equal <code>n.yrs</code>. • <code>\$proj.catch</code> (vector), user-specified aggregate catch for projections. Length must equal <code>n.yrs</code>. • <code>\$avg.yrs</code> (vector), specify which years to average over for calculating reference points. Default = last 5 model years, <code>tail(model\$years, 5)</code>. • <code>\$cont.ecov</code> (T/F), continue ecov process (e.g. random walk or AR1) for projections. Default = TRUE. • <code>\$use.last.ecov</code> (T/F), use terminal year ecov for projections.

	• <code>\$avg.ecov.yrs</code> (vector), specify which years to average over the environmental covariate(s) for projections. • <code>\$proj.ecov</code> (matrix), user-specified environmental covariate(s) for projections. <code>n.yrs x n_Ecov</code>. • <code>\$cont.Mre</code> (T/F), continue M random effects (i.e. AR1_y or 2D AR1) for projections. Default = TRUE. If FALSE, M will be averaged over <code>\$avg.yrs</code> (which defaults to last 5 model years). • <code>\$avg.rec.yrs</code> (vector), specify which years to calculate the CDF of recruitment for use in projections. Default = all model years. • <code>\$percentFXSPR</code> (scalar), percent of <code>F_XSPR</code> to use for calculating catch in projections, only used if <code>proj.opts\$use.FXSPR = TRUE</code>. For example, GOM cod uses <code>F = 75</code> • <code>\$percentFMSY</code> (scalar), percent of <code>F_MSY</code> to use for calculating catch in projections, only used if <code>\$use.FMSY = TRUE</code>.
<code>do.fit</code>	T/F, fit the model using <code>fit_tmb</code> . Default = TRUE.
<code>save.sdrep</code>	T/F, save the full <code>TMB::sdreport</code> object? If FALSE, only save <code>summary.sdreport</code> to reduce model object file size. Default = TRUE.
<code>do.brps</code>	T/F, calculate and report biological reference points. Default = TRUE.
<code>fit.tmb.control</code>	list of optimizer controlling attributes passed to <code>fit_tmb</code> . Default is <code>list(use.optim = FALSE, opt.control = list(iter.max = 1000, eval.max = 1000))</code> , so <code>stats::nlminb</code> is used to optimize.
<code>TMB.bias.correct</code>	T/F whether to use the <code>bias.correct</code> feature of <code>TMB::sdreport</code> . Default = FALSE.
<code>TMB.jointPrecision</code>	T/F whether <code>TMB::sdreport</code> should return the joint precision matrix for the fixed and random effects. Default = FALSE.

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)
```

jitter_wham

Jitter starting values of a fitted WHAM model

Description

Refits a previously fitted WHAM model a number of times with different initial parameter values

Usage

```
jitter_wham(
  fit_RDS = NULL,
  n_jitter = 10,
  initial_vals = NULL,
  which_rows = NULL,
  do_parallel = TRUE,
  n_cores = NULL,
  res_dir = NULL,
  wham_location = NULL,
  test_dir = NULL
)
```

Arguments

<code>fit_RDS</code>	(required) location of RDS file with fitted WHAM model.
<code>n_jitter</code>	the number of different starting locations on the likelihood surface to refit the model from. Default = 10.
<code>initial_vals</code>	(optional) matrix (<code>n_jitter</code> x <code>n</code> parameters) of starting values to use for jittering.
<code>which_rows</code>	(optional) which rows of <code>initial_vals</code> to use for jittering. Useful if doing jitter fits in stages.

do_parallel	T/F whether to do jitter fits in parallel. Requires snowfall and parallel packages to be installed. Default = TRUE.
n_cores	(optional) the number of cores to use for parallel fitting.
res_dir	directory where to save individual files for each jitter refit. Useful if jittering for some reason crashes R. If not provided, no files will be saved.
wham_location	(optional) location of WHAM package. Useful if not using the WHAM installation in the standard library location.
test_dir	(optional) directory for package repository. To be used when the function is being called during package testing rather than an installed version of WHAM.

Details

Because the likelihood surface of a specific model may not be monotone, optimization may result in a local maximum which can be dependent on the parameter values used to start the optimization. It is considered good practice to check whether the optimization is consistent for a range of starting values. There does not appear to be an approach that is both best and computationally efficient to do this check so, this function allows the user to provide the alternative starting values for all fixed effects parameters to use in the jittering. When `initial_vals` is not provided, a default method is used where `n_jitter` random draws from a MVN distribution with mean equal to the MLE from the fitted model and covariance matrix equal to that estimated for the the MLEs.

Value

a list of two elements. First is the results which is a list (length = `n_jitter`) of lists with 3 elements: minimized negative log-likelihood, jitter MLEs, and gradient. Second element is the matrix of initial values for the jitters.

See Also

[fit_wham](#)

make_osa_residuals	<i>Calculate one-step-ahead residuals</i>
--------------------	---

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),
  sdrep_required = 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	<i>Plot WHAM output</i>
------------------	-------------------------

Description

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

Usage

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

Arguments

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

Details

`out.type = 'html'` (default) makes a html file for viewing plot .png files and html tables of parameter estimates, numbers at age, fishing mortality, and spawning stock biomass in a browser. `out.type = 'pdf'` makes one pdf file of all plots and tables. `out.type = 'png'` creates a subdirectory 'plots_png' in `dir.main` and saves .png files within. `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".

`$browse` T/F whether to open the html file in a browser. Default = T.

`$colors` Color palette. Options are "default", "bw", "gray", "cividis", "warm", "ocean", "forest", "highcontrast", "pastel", and "turbo". Default = "default".

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)
```

<code>prepare_projection</code>	<i>Prepare input data and parameters to project an already fit WHAM model</i>
---------------------------------	---

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, check.version = FALSE)
```

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`.
 - `$avg.yrs` (vector), specify which years to use to average population attributes (MAA,FAA,WAA,maturity,movement) in projection years. Any BRPs calculated in projection years will also use these. Default = last 5 years, `tail(model$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.
 - `$avg.ecov.yrs` (vector), specify which years to average the environmental covariate(s) over for projections.
 - `$proj.ecov` (matrix), user-specified environmental covariate(s) for projections. `n.yrs x n.ecov`.
 - `$cont.M.re` (T/F), continue M random effects (i.e. AR1_y or 2D AR1) for projections. Default = FALSE. If FALSE, M will be averaged over `$avg.yrs.M` (which defaults to last 5 model years).
 - `$cont.move.re` (T/F), continue any movement random effects for projections. Default = FALSE. If FALSE, movement parameters will be averaged over `$avg.yrs.move` (which defaults to last 5 model years).
 - `$cont.L.re` (T/F), continue any L ("extra mortality rate") random effects for projections. Default = FALSE. If FALSE, L parameters will be averaged over `$avg.yrs.L` (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 projections, only used if `$use.FXSPR = TRUE`. For example, to project with `F = 75% F_40%SPR`, `proj.opts$percentFXSPR = 75`. Default = 100.
 - `$percentFMSY` (scalar), percent of F_MSY to use for projections, only used if `$use.FMSY = TRUE` and a stock-recruit relationship is assumed. Default = 100.

- `$proj_F_opt` (integer vector), integers specifying how to configure each year of the projection: Overrides any of the above specifications.
 - 1: Use terminal F.
 - 2: use average F (years determined by `$avg.yrs`).
 - 3: use F at X% SPR.
 - 4: user-specified F (from `$proj_Fcatch`).
 - 5: user-specified catch (from `$proj_Fcatch`).
 - 6: use `Fmsy` (if `recruit_model` > 2).
- `$proj_Fcatch` (vector or matrix), catch or F values to use each projection year: values are not used when using `Fmsy`, `FXSPR`, terminal F or average F. Overrides any of the above specifications of `proj.F` or `proj.catch`. if vector, total catch or F is supplied else matrix columns should be fleets for fleet-specific F to be found/used (`n.yrs` x 1 or `n_fleets`).
- `$proj_mature` (array), user-supplied maturity values for the projection years with dimensions (`n_stocks` x `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`).
- `$proj_R_opt` (integer) specifying how recruitment is treated in projection years
 - 1: Continue any RE processes for recruitment.
 - 2: Make projected recruitment consistent with average recruitment in SPR reference points and cancel any bias correction for NAA in projection years.
 - 3: Average recruitment deviations over `$avg.yrs.R` (if `$sigma = "rec"`).
 - 4: No recruitment deviations (if `$sigma = "rec"`).
- `$proj_NAA_opt` (integer) specifying how NAA random effects are treated in projection years if `$sigma = "rec+1"`:
 - 1: Continue any RE processes for NAA.
 - 2: Average NAA deviations over `$avg.yrs.NAA`.
 - 3: No NAA deviations.
- `$proj_NAA_init` (scalar), the default starting value for all NAA random effects in projection years is $\exp(10)$, which may not be large enough for some catch specification. Use this to change the default if a call to `project_wham` suggests it.
- `$proj_F_init` which F to initialize internal newton search for annual projected F for a given user-specified catch. Default is 0.1
- `$avg.yrs.sel` list (length = `n_fleets`), years to average selectivity or FAA for each fleet for projection years. Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.waacatch` list (length = `n_fleets`), years to average weight at age for each fleet for projection years (if `$proj_waa` is NULL). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.

- `$avg.yrs.waassb` list (length = `n_stocks`), years to average weight at age for each stock SSB for projection years (if `$proj_waa` is NULL). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.mature` list (length = `n_stocks`), years to average maturity at age for each stock for projection years (if `$proj_mature` is NULL). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.L` list (length = `n_regions`), years to average extra mortality at age for each region for projection years (if `$cond.L.re = FALSE`). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.M` list (length = `n_stocks`, each is a list with length = `n_regions`), years to average natural mortality at age for each stock and region for projection years (if `$cont.M.re = FALSE`). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.move` list (length = `n_stocks`, each is a list with length = `n_regions`), years to average movement rates at age and season for each stock and region (at beginning of interval) for projection years (if `$cont.move.re = FALSE`). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.R` list (length = `n_stocks`), years to average recruitment deviations for each stock and region for projection years (if `$proj_R_opt = 3`). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.NAA` list (length = `n_stocks`, each is a list with length = `n_regions`), years to average NAA deviations for each stock and region for projection years (if `$proj_NAA_opt = 2`). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.

`check.version` T/F check whether version WHAM and TMB for fitted model match that of the version of WHAM using for projections. Default = TRUE.

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	<i>Prepare input data and parameters for WHAM model</i>
--------------------	---

Description

Prepares data and parameter settings for [fit_wham](#), optionally using an ASAP3 data file read into R by [read_asap3_dat](#). By default, this will set up a SCAA version like **ASAP** without any penalties or random effects. If any asap3, basic_info, catch_info, and index_info arguments are not provided, some respective arbitrary assumptions are made. 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,
  move = NULL,
  L = NULL,
  F = NULL,
  catch_info = NULL,
  index_info = 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 set_ecov for full details)
selectivity	(optional) list specifying selectivity options by block: models, initial values, parameters to fix, and random effects (see set_selectivity for full details)
M	(optional) list specifying natural mortality options: model, random effects, initial values, and parameters to fix (see set_M for full details)

NAA_re	(optional) list specifying options for random effect on numbers-at-age, initial numbers at age, and recruitment model (see set_NAA for full details)
catchability	(optional) list specifying options for priors and random effects on catchability (see set_q for full details)
age_comp	(optional) character or named list, specifies age composition model for fleet(s) and indices (see set_age_comp for full details)
move	(optional) list specifying movement/migration options for models with more than 1 region (see set_move for full details)
L	(optional) list specifying "extra" mortality options (see set_L for full details)
F	(optional) list specifying fishing mortality options (see set_F for full details)
catch_info	(optional) list specifying catch information (see set_catch for full details)
index_info	(optional) list specifying index information (see set_indices for full 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 effects about constant mean
- = 3 Beverton-Holt
- = 4 Ricker

Note: [fit_wham](#) allows 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 models. See [set_ecov](#) for full details.

selectivity specifies options for selectivity, to overwrite existing options specified in the ASAP data file. See [set_selectivity](#) for full details.

M specifies estimation options for natural mortality and can overwrite M-at-age values specified in the ASAP data file. See [set_M](#) for full details.

NAA_re specifies options for random effects on numbers-at-age (NAA, i.e. state-space model or not). See [set_NAA](#) for full details. If NULL, a traditional statistical catch-at-age model is fit.

catchability specifies options for catchability. See [set_q](#) for full details. 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.

move specifies options for movement if there are multiple regions. See [set_move](#) for full details.

age_comp specifies the age composition models for fleet(s) and indices. See [set_age_comp](#) for full details.

`catch_info` is an optional list of fishery catch information that can be used to set up these types of observations when there is no `asap3` file given. See [set_catch](#) for full details. 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.

`index_info` is an optional list of survey/index information that can be used to set up these types of observations when there is no `asap3` file given. See [set_indices](#) for full details. 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.

`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_seasons number of seasons within year.

\$n_fleets number of fishing fleets.

\$fracyr_seasons proportions of year for each season within year (sums to 1).

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

\$waa array ((`n_fleets` + `n_indices` + `n_stocks`) x length(years) x length(ages)) of annual weight at age for each fleet, each index, and spawning biomass for each stock.

\$maturity array (`n_stocks` x length(years) x length(ages)) of annual maturity at age for estimating spawning biomass for each stock.

\$fracyr_SSB matrix (`n_years` x `n_stocks`) (1 or length(years)) of yearly proportions (0-1) of the year at which to calculate spawning biomass.

\$spawn_seasons vector (`n_stocks`) of seasons in which each stock spawns.

\$spawn_regions vector (`n_stocks`) of regions in which each stock spawns.

\$NAA_where array (`n_stocks` x `n_regions` x `n_ages`) of 0/1 indicating where individuals of each stock may exist on January 1 of each year.

\$Fbar_ages integer vector of ages to use to average F at age for reported "Fbar" across all fleets, by regions and by fleet.

\$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

\$XSPR_input_average_years which years to average inputs to per recruit calculation (selectivity, M , WAA , maturity) for static (or prevailing) SPR -based reference points. Default is last 5 years (tail(1:length(years),5)). Below specific values will override this option.

\$average_years_L list (length = `n_regions`), which years to average L (extra mortality) for per recruit calculation for static (or prevailing) reference points. Default is last 5 years (tail(1:length(years),5))

\$average_years_M list (length = `n_stocks`, each is a list with length = `n_regions`), which years to average M for per recruit calculation for static (or prevailing) reference points. Default is last 5 years (tail(1:length(years),5))

- \$average_years_move** list (length = n_stocks, each is a list with length = n_regions), which years to average movement parameters for per recruit calculation for static (or prevailing) reference points. Default is last 5 years (tail(1:length(years),5))
- \$average_years_mat** list (length = n_stocks), which years to average maturity for per recruit calculation for static (or prevailing) reference points. Default is last 5 years (tail(1:length(years),5))
- \$average_years_waassb** list (length = n_stocks), which years to average ssb waa for per recruit calculation for static (or prevailing) reference points. Default is last 5 years (tail(1:length(years),5))
- \$average_years_sel** list (length = n_fleets), which years to average selectivity for per recruit calculation for static (or prevailing) reference points. Default is last 5 years (tail(1:length(years),5))
- \$average_years_waacatch** list (length = n_fleets), which years to average catch waa for per recruit calculation for static (or prevailing) reference points. Default is last 5 years (tail(1:length(years),5))
- \$average_years_SRR** list (length = n_stocks), which years to average SRR parameters for static (or prevailing) MSY-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. For long-term projections, may be important to use certain years for XSPR_R_avg_yrs
- \$FXSPR_init** which F to initialize internal newton search for annual and static F at X percent SPR. Default is 0.5
- \$FMSY_init** which F to initialize internal newton search for annual and static Fmsy (if a stock-recruit model is assumed). Default is 0.5
- \$simulate_process_error** T/F vector (length = 9). When simulating from the model, whether to simulate any process errors for (NAA, M, selectivity, q, movement, unidentified mortality, q priors, movement priors, Ecov). Only used for applicable random effects.
- \$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_BRPs** T/F. Perform bias correction of analytic SSB/R and Y/R when there is bias correction of log-normal NAA. May want to use XSPR_R_opt = 5 for long-term projections.

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)
 log list of character strings attempting to describe the input and what the model assumes.

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, cont.ecov = TRUE, use.last.ecov = FALSE, percentFXSPR = 100,
    percentFMSY = 100),
  n.newton = 3,
  do.sdrep = TRUE,
  MakeADFun.silent = FALSE,
  save.sdrep = TRUE,
  check.version = TRUE,
```

```

    TMB.bias.correct = FALSE,
    TMB.jointPrecision = FALSE
  )

```

Arguments

- | | |
|-----------|---|
| model | a previously fit wham model |
| proj.opts | a named list with the following components: <ul style="list-style-type: none"> • <code>\$n.yrs</code> (integer), number of years to project/forecast. Default = 3. • <code>\$use.last.F</code> (T/F), use terminal year F for projections. Default = TRUE. • <code>\$use.avg.F</code> (T/F), use average of F over certain years for projections. Default = FALSE. Years to average over determined by <code>\$avg.yrs</code> defined below. • <code>\$use.FXSPR</code> (T/F), calculate and use F at X% SPR for projections. Default = FALSE. • <code>\$use.FMSY</code> (T/F), calculate and use FMSY for projections. Default = FALSE. • <code>\$proj.F</code> (vector), user-specified fishing mortality for projections. Length must equal <code>n.yrs</code>. • <code>\$proj.catch</code> (vector), user-specified aggregate catch for projections. Length must equal <code>n.yrs</code>. • <code>\$avg.yrs</code> (vector), specify which years to use to average population attributes (MAA,FAA,WAA,maturity,movement) in projection years. Any BRPs calculated in projection years will also use these. Default = last 5 years, <code>tail(model\$years, 5)</code>. • <code>\$cont.ecov</code> (T/F), continue ecov process (e.g. random walk or AR1) for projections. Default = TRUE. • <code>\$use.last.ecov</code> (T/F), use terminal year ecov for projections. • <code>\$avg.ecov.yrs</code> (vector), specify which years to average the environmental covariate(s) over for projections. • <code>\$proj.ecov</code> (matrix), user-specified environmental covariate(s) for projections. <code>n.yrs x n.ecov</code>. • <code>\$cont.M.re</code> (T/F), continue M random effects (i.e. AR1_y or 2D AR1) for projections. Default = FALSE. If FALSE, M will be averaged over <code>\$avg.yrs.M</code> (which defaults to last 5 model years). • <code>\$cont.move.re</code> (T/F), continue any movement random effects for projections. Default = FALSE. If FALSE, movement parameters will be averaged over <code>\$avg.yrs.move</code> (which defaults to last 5 model years). • <code>\$cont.L.re</code> (T/F), continue any L ("extra mortality rate") random effects for projections. Default = FALSE. If FALSE, L parameters will be averaged over <code>\$avg.yrs.L</code> (which defaults to last 5 model years). • <code>\$avg.rec.yrs</code> (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). • <code>\$percentFXSPR</code> (scalar), percent of F_XSPR to use for projections, only used if <code>\$use.FXSPR = TRUE</code>. For example, to project with F = 75% F_40%SPR, <code>proj.opts\$percentFXSPR = 75</code>. Default = 100. |

- `$percentFMSY` (scalar), percent of `F_MSY` to use for projections, only used if `$use.FMSY = TRUE` and a stock-recruit relationship is assumed. Default = 100.
- `$proj_F_opt` (integer vector), integers specifying how to configure each year of the projection: Overrides any of the above specifications.
 - 1: Use terminal F.
 - 2: use average F (years determined by `$avg.yrs`).
 - 3: use F at X% SPR.
 - 4: user-specified F (from `$proj_Fcatch`).
 - 5: user-specified catch (from `$proj_Fcatch`).
 - 6: use `Fmsy` (if `recruit_model > 2`).
- `$proj_Fcatch` (vector or matrix), catch or F values to use each projection year: values are not used when using `Fmsy`, `FXSPR`, terminal F or average F. Overrides any of the above specifications of `proj.F` or `proj.catch`. if vector, total catch or F is supplied else matrix columns should be fleets for fleet-specific F to be found/used (`n.yrs x 1` or `n_fleets`).
- `$proj_mature` (array), user-supplied maturity values for the projection years with dimensions (`n_stocks x 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`).
- `$proj_R_opt` (integer) specifying how recruitment is treated in projection years
 - 1: Continue any RE processes for recruitment.
 - 2: Make projected recruitment consistent with average recruitment in SPR reference points and cancel any bias correction for NAA in projection years.
 - 3: Average recruitment deviations over `$avg.yrs.R` (if `$sigma = "rec"`).
 - 4: No recruitment deviations (if `$sigma = "rec"`).
- `$proj_NAA_opt` (integer) specifying how NAA random effects are treated in projection years if `$sigma = "rec+1"`:
 - 1: Continue any RE processes for NAA.
 - 2: Average NAA deviations over `$avg.yrs.NAA`.
 - 3: No NAA deviations.
- `$proj_NAA_init` (scalar), the default starting value for all NAA random effects in projection years is `exp(10)`, which may not be large enough for some catch specification. Use this to change the default if a call to `project_wham` suggests it.
- `$proj_F_init` which F to initialize internal newton search for annual projected F for a given user-specified catch. Default is 0.1
- `$avg.yrs.sel` list (length = `n_fleets`), years to average selectivity or FAA for each fleet for projection years. Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.waacatch` list (length = `n_fleets`), years to average weight at age for each fleet for projection years (if `$proj_waa` is NULL). Any BRPs

calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.

- `$avg.yrs.waassb` list (length = `n_stocks`), years to average weight at age for each stock SSB for projection years (if `$proj_waa` is NULL). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.mature` list (length = `n_stocks`), years to average maturity at age for each stock for projection years (if `$proj_mature` is NULL). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.L` list (length = `n_regions`), years to average extra mortality at age for each region for projection years (if `$cond.L.re` = FALSE). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.M` list (length = `n_stocks`, each is a list with length = `n_regions`), years to average natural mortality at age for each stock and region for projection years (if `$cont.M.re` = FALSE). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.move` list (length = `n_stocks`, each is a list with length = `n_regions`), years to average movement rates at age and season for each stock and region (at beginning of interval) for projection years (if `$cont.move.re` = FALSE). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.R` list (length = `n_stocks`), years to average recruitment deviations for each stock and region for projection years (if `$proj_R_opt` = 3). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.
- `$avg.yrs.NAA` list (length = `n_stocks`, each is a list with length = `n_regions`), years to average NAA deviations for each stock and region for projection years (if `$proj_NAA_opt` = 2). Any BRPs calculated in projection years will also use this. Default = last 5 years, `tail(model$years, 5)`.

<code>n.newton</code>	integer, number of additional Newton steps after optimization. Passed to <code>fit_tmb</code> . Default = 0 for projections.
<code>do.sdrep</code>	T/F, calculate standard deviations of model parameters? See <code>sdreport</code> . Default = TRUE.
<code>MakeADFun.silent</code>	T/F, Passed to silent argument of <code>TMB::MakeADFun</code> . Default = FALSE.
<code>save.sdrep</code>	T/F, save the full <code>TMB::sdreport</code> object? If FALSE, only save <code>summary.sdreport</code> to reduce model object file size. Default = TRUE.
<code>check.version</code>	T/F check whether version WHAM and TMB for fitted model match that of the version of WHAM using for projections. Default = TRUE.
<code>TMB.bias.correct</code>	T/F whether to use the <code>bias.correct</code> feature of <code>TMB::sdreport</code> . Default = FALSE.
<code>TMB.jointPrecision</code>	T/F whether to return the joint precision matrix for the fixed and random effects from <code>TMB::sdreport</code> . Default = FALSE.

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. $\log(R) \sim N(\text{meanlogR}, \text{sdlogR})$.

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 quantities 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 vector, names of ASAP3 .dat files. The file either needs to be in the current working directory, or filename can include the path. If multiple files, a multi-stock model will be assumed.

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	<i>Read ASAP3 fit</i>
----------------	-----------------------

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*SS-BPR(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	<i>Read WHAM fit</i>
---------------	----------------------

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 array, stocks x regions x 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_rep array, stocks x regions x years x ages, numbers at age
- \$NAA_CV array, stocks x regions x 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](#)

reduce_input	<i>Reduce the years of the model</i>
--------------	--------------------------------------

Description

Internal function called by `fit_peel` for i in $1:n.peels$. Creates the input for the model peeling off i years of data (calls `fit_tmb`). Reduces the year dimension of all data, parameters, maps, so that one can project correctly from the peeled model.

Usage

```
reduce_input(input, years_peeled, retro = TRUE)
```

Arguments

input	list containing data, parameters, map, and random elements (output from <code>prepare_wham_input</code>). NOT from <code>prepare_projection</code> or <code>project_wham</code>
years_peeled	which of <code>input\$years</code> to peel from the model input
retro	(T/F) whether this is for a retro peel (Default = TRUE)

retro	<i>Run retrospective analysis</i>
-------	-----------------------------------

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 = NULL,
  use.mle = TRUE,
  do.sdrep = FALSE,
  n.newton = 0,
  MakeADFun.silent = FALSE,
  retro.silent = FALSE,
  save.input = FALSE,
  do.brps = FALSE,
  check.version = TRUE,
  save.sdrep = 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 = "model\$input\$random".
use.mle	T/F, use MLEs from full model fit as initial values for each peel? If not, the initial values from full model input are used. Default = TRUE.
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.
do.brps	T/F, calculate and report biological reference points
check.version	T/F, whether to verify the wham package commit and version for the fitted model are the same as the currently used package.
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

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](#)

self_test

Perform self-test simulation and estimation of a fitted WHAM model

Description

Generate simulated data from fitted model and refit the model to the simulated data

Usage

```
self_test(
  fit_RDS = NULL,
  n = 10,
  seeds = NULL,
  which_seeds = NULL,
  conditional = TRUE,
  map_change = NULL,
  do_parallel = TRUE,
  n_cores = NULL,
  res_dir = NULL,
  wham_location = NULL,
  test_dir = NULL,
  save_inputs = FALSE
)
```

Arguments

fit_RDS	(required) location of RDS file with fitted WHAM model.
n	the number of simulated data sets to create and fit. Default = 10.
seeds	(optional) vector of seeds to use to generate each simulated data set.
which_seeds	(optional) which seeds to use for simulated data set. Useful if doing self test in stages.
conditional	T/F whether to fix rather than simulate estimated random effects. Default = TRUE.
map_change	(optional) list of input\$map elements for altering mapping assumptions of fitted model.
do_parallel	T/F whether to do self-test fits in parallel. Requires snowfall and parallel packages to be installed. Default = TRUE.
n_cores	(optional) the number of cores to use for parallel fitting.
res_dir	directory where to save individual files for each self test fit. Useful if doing self test in stages. If not provided, no files will be saved.
wham_location	(optional) location of WHAM package. Useful if not using the WHAM installation in the standard library location.
test_dir	(optional) directory for package repository. To be used when the function is being called during package testing rather than an installed version of WHAM.
save_inputs	T/F whether to save the simulated inputs in res_dir. Default = FALSE.

Value

a list of two elements. First is the results which is a list (length = n) of lists with 6 elements: minimized negative log-likelihood, MLEs, gradient, SSB, F, abundance at age. Second element is the vector of seeds used for self-test simulations.

See Also[fit_wham](#)

set_age_comp

*Specify the age composition models for fleet(s) and indices.***Description**

Specify the age composition models for fleet(s) and indices.

Usage

```
set_age_comp(input, age_comp)
```

Arguments

- | | |
|----------|---|
| input | list containing data, parameters, map, and random elements (output from wham::prepare_wham_input) |
| age_comp | <p>specifies the age composition models for fleet(s) and indices. If NULL, the multinomial is used because this was the only option in ASAP.</p> <p>The age composition models available are:</p> <ul style="list-style-type: none"> "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 options 8-10 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.

Value

a named list with same elements as the input provided with age composition likelihood options modified.

See Also

[prepare_wham_input](#)

Examples

```
## Not run:
wham.dir <- find.package("wham")
path_to_examples <- system.file("extdata", package="wham")
asap3 <- read_asap3_dat(file.path(path_to_examples,"ex1_SNEMAYT.dat"))
input <- prepare_wham_input(asap3)
input <- set_age_comp(input, age_comp = "logistic-normal-miss0") #no longer multinomial

## End(Not run)
```

set_catch	<i>Specify catch selectivity blocks and aggregate and age composition observations for catch</i>
-----------	--

Description

Specify catch selectivity blocks and aggregate and age composition observations for catch

Usage

```
set_catch(input, catch_info = NULL)
```

Arguments

input	list containing data, parameters, map, and random elements (output from prepare_wham_input)
catch_info	(optional) list specifying various aspects about catch by fleet (see details) catch_info specifies observations, and various configuration options for fleet-specific catch observations and will overwrite attributes specified in the ASAP data file. If NULL, all settings from the ASAP data file or basic_info are used. catch_info is a list with any of the following entries: \$n_fleets number of fleets \$fleet_regions vector (n_fleets) of regions where each fleet operates. \$fleet_seasons matrix (n_fleets x n_seasons) of 0/1 values flagging which seasons each fleet operates. \$fleet_names character vector (n_fleets) of names for fleets. Used for naming results in plots and tables. \$agg_catch matrix (n_years_model x n_fleets) of annual aggregate catches by fleet. \$agg_catch_cv matrix (n_years_model x n_fleets) of CVs for annual aggregate catches by fleet. \$scatch_paa array (n_fleets x n_years_model x n_ages) of annual catch proportions at age by fleet. \$use_catch_paa matrix (n_years_model x n_fleets) of 0/1 values flagging whether to use proportions at age observations. \$scatch_Neff matrix (n_years_model x n_fleets) of effective sample sizes for proportions at age observations. \$waa_pointer_fleets vector (n_fleets) of itegers indicated waa to use for each fleet. \$selblock_pointer_fleets matrix (n_years_model x n_fleets) of itegers indicated selblocks to use. \$initial_catch_sd_scale vector (n_fleets) of scalar multipliers of annual log-observation standard deviation. Default = 1. \$map_catch_sd_scale integer vector (n_fleets) specifying which sd scalar parameters to fix. Use NA to fix a parameter and integers to estimate. Use the same integer for multiple fleets to estimate a shared scalar parameter.

Value

a named list with same elements as the input provided with catch observations and fleet options modified.

See Also

[prepare_wham_input](#)

Examples

```
## Not run:
wham.dir <- find.package("wham")
path_to_examples <- system.file("extdata", package="wham")
asap3 <- read_asap3_dat(file.path(path_to_examples,"ex1_SNEMAYT.dat"))
input <- prepare_wham_input(asap3)
input <- set_catch(input, catch_info = list(agg_catch = newcatch)) #constant catch of 500 mt

## End(Not run)
```

set_ecov	<i>Specify configuration for environmental covariates, effects on the population, and parameter values</i>
----------	--

Description

Specify configuration for environmental covariates, effects on the population, and parameter values

Usage

```
set_ecov(input, ecov)
```

Arguments

- input list containing data, parameters, map, and random elements (output from [wham::prepare_wham_input](#))
- ecov (optional) named list of environmental covariate data and parameters (see details)
ecov specifies any environmental covariate data and model as well as the effect on the population. 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 1/0) 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.

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

\$process_mean_vals vector of (initial) mean values for the ecov time-series.

\$process_sig_vals vector of (initial) standard deviation values for the ecov time-series.

\$process_cor_vals vector of (initial) correlation values for the ecov time-series.

\$recruitment_how character matrix (n_Ecov x n_stocks) indicating how each ecov affects recruitment for each stock. Options are based on (see [Iles & Beverton \(1998\)](#)) combined with the order of orthogonal polynomial of the covariate and has the form "type-lag-order". "type" can be:

= "none" no effect.

= "controlling" pre-recruit density-independent mortality.

= "limiting" maximum recruitment, e.g. ecov determines amount of suitable habitat)

= "lethal" threshold, i.e. $R \rightarrow 0$ at some ecov value.

= "masking" metabolic/growth, decreases dR/dS

= "directive" e.g. behavioral

for type other than "none", "lag" can be:

= "lag-n" lag = n which can be 0,1,2,... lag-1 implies the covariate in year y affects recruitment in year y+1.

for "type" being other than "none", "order" can be:

= "linear" the covariate effect is linear on the transformed recruitment parameter (e.g., log).

= "poly-n" orthogonal polynomial where n = 1 (same as "linear"),2,...

so "limiting-lag-1-poly-2" would model the covariate affecting recruitment the next year (lag = 1) as a second order orthogonal polynomial to represent $(b_0 + b_1 * Ecov + b_2 * Ecov + \dots)$ limiting effect.

\$M_how character array (n_Ecov x n_stocks x n_ages x n_regions) indicating how each ecov affects M by age,stock,region and has the form "lag-order". "lag" can be:

= **"none"** no effect.

= **"lag-n"** lag = n which can be 0,1,2,... lag-1 implies the covariate in year y affects M in year y+1.

for "lag" being other than "none", "order" can be:

= **"linear"** the covariate effect is linear on the transformed M parameter (e.g., log).

= **"poly-n"** orthogonal polynomial where n = 1 (same as "linear"),2,...

\$M_effect_map integer array (n_stocks x n_ages x n_regions x n_Ecov) indicating which estimated effects are common by age,stock,region. If not specified there the same effect is estimated for all M where \$M_how is other than "none" for each covariate.

\$q_how character matrix (n_Ecov x n_indices) indicating whether each ecov affects catchability for each index. and has the form "lag-order". "lag" can be:

= **"none"** no effect.

= **"lag-n"** lag = n which can be 0,1,2,... lag-1 implies the covariate in year y affects catchability in year y+1.

for "lag" being other than "none", "order" can be:

= **"linear"** the covariate effect is linear on the transformed catchability parameter (e.g., log).

= **"poly-n"** orthogonal polynomial where n = 1 (same as "linear"),2,...

\$move_how character array (n_Ecov x n_stocks x n_ages x n_seasons x n_regions x n_regions - 1) indicating whether each ecov affects movement from one region to the others by stock,age,season. and has the form "lag-order". "lag" can be:

= **"none"** no effect.

= **"lag-n"** lag = n which can be 0,1,2,... lag-1 implies the covariate in year y affects a movement parameter in year y+1.

for "lag" being other than "none", "order" can be:

= **"linear"** the covariate effect is linear on the transformed movement parameter (e.g., log).

= **"poly-n"** orthogonal polynomial where n = 1 (same as "linear"),2,...

\$move_effect_map integer array (n_stocks x n_ages x n_seasons x n_regions x n_regions-1 x n_Ecov) indicating which estimated effects are common by age,stock,region, season etc. If not specified the same effect is estimated for all movement parameters where \$move_how is other than "none" for each covariate.

\$beta_R_vals n_stocks x n_ecov x max(n_poly_R) array of initial values for effects on recruitment.

\$beta_M_vals n_stocks x n_ages x n_regions x n_ecov x max(n_poly_M) array of initial values for effects on natural mortality.

\$beta_q_vals n_indices x n_ecov x max(n_poly_q) array of initial values for effects on catchability.

\$beta_mu_vals n_stocks x n_ages x n_seasons x n_regions x n_regions - 1 x n_ecov x max(n_poly_move) array of initial values for effects on movement parameters.

Value

a named list with same elements as the input provided with environmental covariate observations, effects, and model options modified.

See Also

[prepare_wham_input](#)

Examples

```
## Not run:
wham.dir <- find.package("wham")
path_to_examples <- system.file("extdata", package="wham")
asap3 <- read_asap3_dat(file.path(path_to_examples,"ex1_SNEMAYT.dat"))
env.dat <- read.csv(file.path(path_to_examples,"GSI.csv"), header=T)
input <- prepare_wham_input(asap3, NAA_re = list(sigma = "rec"))
ecov <- list(
  label = "GSI",
  mean = as.matrix(env.dat$GSI),
  logsigma = 'est_1', # estimate obs sigma, 1 value shared across years
  year = env.dat$year,
  use_obs = matrix(1, ncol=1, nrow=dim(env.dat)[1]), # use all obs (=1)
  process_model = 'ar1') # "rw" or "ar1"
input <- set_ecov(input, ecov = ecov) #GSI in the model without any effects

## End(Not run)
```

set_F	<i>Specify configuration for fully-selected fishing mortality</i>
-------	---

Description

Specify configuration for fully-selected fishing mortality

Usage

```
set_F(input, F_opts = NULL)
```

Arguments

- input list containing data, parameters, map, and random elements (output from [wham::prepare_wham_input](#))
- F_opts (optional) named list of initial values for annual fully-selected fishing mortality and configuration method for estimation.

F_opts specifies a few as well as the effect on the population. 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:

\$F matrix (n_years x n_fleets) of (initial) values for fully-selected fishing mortality.

\$F_config integer 1: (default) configure F parameters (on log scale) as an F in the initial year and then deviations from one year to the next, or 2: configure F parameters as (log) annual values.

\$map_F Specify whether to fix any fully-selected F parameters, corresponds to map\$F_pars. integer matrix (n_years x n_fleets). Use NA to fix parameters and common integers will make those parameters equal. E.g., for a given column (fleet) values NA,1,2,... will fix the first year and estimate other years. Use unique values for all distinct parameters for each year and fleet.

set_indices	<i>Specify index selectivity blocks and aggregate and age composition observations for indices</i>
-------------	--

Description

Specify index selectivity blocks and aggregate and age composition observations for indices

Usage

```
set_indices(input, index_info = NULL)
```

Arguments

input	list containing data, parameters, map, and random elements (output from prepare_wham_input)
index_info	(optional) list specifying various aspects about catch by indices (see details) index_info specifies observations, and various configuration options for index-specific catch observations and will overwrite attributes specified in the ASAP data file. If NULL, all settings from the ASAP data file or basic_info are used. index_info is a list with any of the following entries: \$n_indices number of indices \$index_regions vector (n_indices) of regions where each fleet operates. \$index_seasons vector (n_indices) of 0/1 values flagging which seasons each index occurs. \$index_names character vector (n_indices) of names for indices Used for naming results in plots and tables. \$agg_indices matrix (n_years_model x n_indices) of annual aggregate index catches. \$agg_index_cv matrix (n_years_model x n_indices) of CVs for annual aggregate index catches. \$fracyr_indices matrix (n_years_model x n_indices) of fractions of year at which index occurs within the season (difference between time of survey and time at start of season).

\$use_indices matrix (n_years_model x n_indices) of 0/1 values flagging whether to use aggregate observations.

\$units_indices matrix (n_years_model x n_indices) of 1/2 values flagging whether aggregate observations are biomass (1) or numbers (2).

\$index_paa array (n_indices x n_years_model x n_ages) of annual catch proportions at age by index.

\$use_index_paa matrix (n_years_model x n_indices) of 0/1 values flagging whether to use proportions at age observations.

\$units_index_paa matrix (n_years_model x n_indices) of 1/2 values flagging whether composition observations are biomass (1) or numbers (2).

\$index_Neff matrix (n_years_model x n_indices) of effective sample sizes for proportions at age observations.

\$waa_pointer_indices vector (n_indices) of itegers indicated waa to use for each index.

\$selblock_pointer_indices matrix (n_years_model x n_indices) of itegers indicated selblocks to use.

\$initial_index_sd_scale vector (n_indices) of scalar multipliers of annual log-observation standard deviation. Default = 1.

\$map_index_sd_scale integer vector (n_indices) specifying which sd scalar parameters to fix. Use NA to fix a parameter and integers to estimate. Use the same integer for multiple indices to estimate a shared scalar parameter.

set_L

Specify model and parameter configuration for "extra" mortality not directly attributed to natural mortality

Description

Specify model and parameter configuration for "extra" mortality not directly attributed to natural mortality

Usage

```
set_L(input, L)
```

Arguments

input list containing data, parameters, map, and random elements (output from [wham::prepare_wham_input](#))

L (optional) list specifying "extra" mortality options: model, random effects, initial values, and parameters to fix (see details)

L specifies estimation options for "extra" mortality. If NULL, This mortality source is not used. L is a list with the following entries:

\$model length = n_regions. "extra" mortality model options are:

"none" (default) no extra mortality for this region.

"**constant**" estimate a single mean mortality for the region shared across all ages

"**iid_re**" estimate independent random effects over years, for the region

"**ar1_re**" estimate random effect correlated over years, for the region

\$initial_means Initial/mean L-at-region

\$sigma_vals Initial standard deviation by region value to use for the L random effects. Values are not used if `L$model = "none"`.

\$cor_vals Initial correlation values to use for the L random effects. If unspecified all initial values are 0

set_M

Specify model and parameter configuration for natural mortality

Description

Specify model and parameter configuration for natural mortality

Usage

```
set_M(input, M)
```

Arguments

input list containing data, parameters, map, and random elements (output from `wham::prepare_wham_input`)

M (optional) list specifying natural mortality options: model, random effects, initial values, and parameters to fix (see details)

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 `M$means_map` (see [vignette](#) for more details). M is a list with the following entries:

\$mean_model Character describing the type of model for M stock and regional models for natural mortality. Options are:

"**fixed-M**" Use initial values from ASAP3 dat files or `$initial_means` for (mean) M as fixed values. If no ASAP3 files and `$initial_means` is not provided, default is `M = 0.2` for all stocks, regions and ages

"**estimate-M**" estimate one or more (mean) M parameters. Default is to estimate a single M shared across all stocks and ages, but use `$means_map` to fix or estimate parameters for specific stocks, regions, ages.

"**weight-at-age**" specifies M as a function of weight-at-age, $M_{y,a} = \exp(b_0 + b_1 * \log(W_{y,a}))$, as in [Lorenzen \(1996\)](#) and [Miller & Hyun \(2018\)](#). Default is to estimate a single model shared across all stocks and regions, but use `$means_map[s, r, 1]` to fix or estimate the intercept for specific stocks, regions. See also `$logb_prior` and `$initial_b` configuring the slope on log scale.

- \$initial_means** array (n_stocks x n_regions x n_ages) of initial/mean M by stock, region and age. If NULL, initial mean M-at-age values for a given stock and region are taken from the first row of the MAA matrix in the ASAP data file. If no ASAP data file, $M = 0.2$ is the default. If \$mean_model is "weight-at-age" only elements for the first age (\$initial_means[, , 1]) are used (for the intercept of $\log(M)$).
- \$means_map** array (n_stocks x n_regions x n_ages) of NA or integers ($0 \leq \max \leq n_stocks * n_regions * n_ages$) indicating which ages to estimate (mean) M and whether to set any ages to be identical. E.g. in a model with 2 stock, 2 regions and 6 ages with constant M estimated for each stock across regions and ages \$M_ages_map[1, ,] = 1 and \$M_ages_map[2, ,] = 2. \$M_ages_map[1, 1,] = c(NA, 1, 1, 2, 2, 3) will fix M for age 1 at the initial value, and estimates for ages 2 and 3 are identical as are those for ages 4 and 5 and different from age 6+ for stock 1 and region 1. If NULL, specifies all ages fixed at M\$initial_means. If \$mean_model is "weight-at-age" these are used for all stocks and regions and only the elements for the first age (\$M_ages_map[, , 1]) are used (for the intercept of $\log(M)$).
- \$intial_MAA** array (n_stocks x n_regions x n_years x n_ages) of initial values for M at age. Intended to be uses when nothing pertaining to M estimated.
- \$b_model** "constant", "stock", "region", "stock-region" defining whether parameter is constant, stock-specific, region-specific, stock- and region-specific. Only used if M\$mean_model = "weight-at-age".
- \$b_prior** T/F, should a $N(\mu, 0.08)$ prior (where $\mu = \log(0.305)$ by default) be used on $\log_b$? Based on Fig. 1 and Table 1 (marine fish) in [Lorenzen \(1996\)](#). (Only used if \$mean_model is "weight-at-age").
- \$intial_b** if any elements of \$mean_model is "weight-at-age", initial value for mean b for weight-at-age model.
- \$re_model** Character matrix (n_stocks x n_regions) of options for time- and age-varying (random effects) on M by stock and region. Possible values are:
- "none" (default) No random effects by age or year.
 - "iid_a" uncorrelated M by age, constant in time.
 - "iid_y" uncorrelated M by year, constant all ages.
 - "ar1_a" M correlated by age (AR1), constant in time.
 - "ar1_y" M correlated by year (AR1), constant all ages.
 - "iid_ay" M uncorrelated by year and age (2D).
 - "ar1_ay" M correlated by year and age (2D AR1), as in [Cadigan \(2016\)](#).
- \$re_map** array (n_stocks x n_regions x n_ages) of NA and integers ($1 \leq \max \leq n_ages$) indicating which ages, for a given stock and region, have random effects (not NA) and whether to set RE for any ages to be identical. E.g. in a model with 2 stock, 2 regions and 6 ages, \$re_map[2, 1,] = c(NA, 1, 1, 2, 2, 3) will not estimate RE for age 1, and those for ages 2 and 3 are identical as are those for ages 4 and 5 and different from age 6+ for stock 2 and region 1. If NULL, and \$re_model specifies M random effects at age, at least two ages must be specified for correlation among ages to be estimated.

\$sigma_vals n_stocks x n_regions matrix Initial standard deviation value to use for the M random effects. Values are not used if M\$re_model = "none". Otherwise, a single value. If unspecified all values are 0.1.

\$cor_vals n_stocks x n_regions x 2 array of initial correlation values to use for the M deviations. If unspecified all initial values are 0. When M\$re_model = "iid_a", "iid_y", "iid_ay" or "none" values are not used. "ar1_a" first value cor_vals[s,r,1] is used. "ar1_y" second value cor_vals[s,r,2] is used. "ar1_ay" First is for "age", second is for "year".

\$sigma_map n_stocks x n_region matrix of NA or integers indicating which random effects sd is estimated and whether to set any to be identical. If not supplied a single sd will be estimated for any stock and region where \$re_model is other than "none".

\$cor_map n_stocks x n_region matrix x 2 array of NA or integers indicating which random effects correlation parameters are estimated and whether to set any to be identical. If not supplied a single value for age and/or year will be estimated for any stock and region where \$re_model is other than "none", "iid_a", "iid_y".

Value

a named list with same elements as the input provided with natural mortality options modified.

See Also

[prepare_wham_input](#)

Examples

```
## Not run:
wham.dir <- find.package("wham")
path_to_examples <- system.file("extdata", package="wham")
asap3 <- read_asap3_dat(file.path(path_to_examples,"ex1_SNEMAYT.dat"))
input <- prepare_wham_input(asap3)
M = list(mean_model = "estimate-M")
input <- set_M(input, M = M) #estimate a constant M parameters

## End(Not run)
```

set_move

Specify model and parameter configuration for movement when input\$data\$n_regions > 1

Description

Specify model and parameter configuration for movement when input\$data\$n_regions > 1

Usage

```
set_move(input, move)
```

Arguments

input list containing data, parameters, map, and random elements (output from `wham::prepare_wham_input`)

move (optional) list specifying movement options: model, random effects, initial values, and parameters to fix (see details)

move specifies estimation options for movement. If NULL, no movement will occur. If there are multiple regions, each stock will be modeled separately in different regions without movement. **move** is a list with the following entries:

\$stock_move length = `n_stocks`, T/F whether each stock can move. If not provided then movement will be defined below for all stocks.

\$separable length = `n_stocks`, T/F whether movement should be modeled separately from mortality or both occurring simultaneously.

\$mean_model matrix (`n_regions` x (`n_regions`-1)): model options for fixed effects (mean and possibly variance) for each movement parameter are:

- "none" (default) no movement between regions.
- "constant" estimate a single movement rate to each region shared across all stocks, seasons, ages, years
- "season" estimate movement rates to each region for each season shared across all stocks, ages, years
- "stock_constant" estimate a movement rate for each stock to each region shared across all seasons, ages, years
- "stock_season" estimate a movement rate for each stock each season to each region shared across all ages, years

\$age_re matrix (`n_regions` x (`n_regions`-1)): options for age random effects (for each mean parameter defined in `move$mean_model`):

- "none" (default) no movement rate random effects by age.
- "iid" independent movement rate random effects by age.
- "ar1" allow first order autoregressive correlation of movement rate random effects by age.

\$year_re matrix (`n_regions` x (`n_regions`-1)): options for yearly random effects (for each mean parameter defined in `move$mean_model`):

- "none" (default) no movement rate random effects by year.
- "iid" independent movement rate random effects by year.
- "ar1" allow first order autoregressive correlation of movement rate random effects by year.

\$prior_sigma array (`n_stocks` x `n_seasons` x `n_regions` x `n_regions` - 1) of sd parameters for normal priors on mean movement parameters on transformed scale (-Inf,Inf)

\$use_prior array (`n_stocks` x `n_seasons` x `n_regions` x `n_regions` - 1) 0/1 indicator whether to include prior for mean movement parameters in joint log-likelihood.

\$scan_move array (n_stocks x n_seasons x n_regions x n_regions) 0/1 indicator whether movement can occur from one region to another.

\$must_move array (n_stocks x n_seasons x n_regions) 0/1 indicator whether movement from region must occur.

\$mean_vals array (n_stocks x n_seasons x n_regions x n_regions-1) of initial movement rate parameters *from* each region. Usage depends on move\$mean_model.

\$sigma_vals array (n_stocks x n_seasons x n_regions x n_regions -1) of initial standard deviations to use for random effects. Usage depends on move\$age_re and move\$year_re.

\$cor_vals array (n_stocks x n_seasons x n_regions x n_regions - 1x 2) of initial correlation values to use for random effects. Usage depends on move\$age_re and move\$year_re. cor_vals[,,,1] is for correlation with age, and cor_vals[,,,2] is for correlation with year.

set_NAA	<i>Specify model and parameter configuration for numbers at age</i>
---------	---

Description

Specify model and parameter configuration for numbers at age

Usage

```
set_NAA(input, NAA_re = NULL)
```

Arguments

input list containing data, parameters, map, and random elements (output from [prepare_wham_input](#))

NAA_re (optional) list specifying options for numbers-at-age random effects, initial parameter values, and recruitment model (see details)

If NAA_re = NULL, a traditional statistical catch-at-age model is fit (NAA = pred_NAA for all ages, deterministic). Otherwise, NAA_re specifies numbers-at-age configuration. It is a list with the following possible entries:

\$decouple_recruitment T/F determining whether correlation structure of recruitment is independent of RE deviations for older ages (default = TRUE). 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 "ar1_y" or "2dar1" separate year correlation parameters are estimated for recruitment and older ages.

\$sigma Which ages allow deviations from the predicted NAA given NAA from previous time step? Must be a single character string described below or a vector of length n_stocks. If length = 1, assumptions will be applied to all stocks. 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

\$sigma_vals Array (`n_stocks` x `n_regions` x `n_ages`) of initial standard deviation values to use for the NAA deviations. Values are not used if `recruit_model = 1` and `NAA_re$sigma` is not specified. Only those for age 1 in the spawning region are used if `NAA_re$sigma = "rec"`. If `NAA_re$sigma_map` is defined, the user must ensure that the configuration is compatible with `NAA_re$sigma_vals`

\$sigma_map You can specify a more complex parameter configuration by entering an integer array (`n_stocks` x `n_regions` x `n_ages`), where each entry points to the `NAA_sigma` to use for that stock and age. E.g., for 2 stocks, 2 regions, and 6 ages, `array(rep(c(1,2,2,3,3,3), each = 4), c(2,2,6))` will estimate 3 sigmas, with recruitment (age-1) deviations having their own `sigma_R`, ages 2-3 sharing `sigma_2`, and ages 4-6 sharing `sigma_3`. All parameters being the same for both stocks and across regions. The user must be sure that a compatible `NAA_re$sigma` configuration is defined. Values are not used if `recruit_model = 1` and `NAA_re$sigma` is not specified.

\$cor Correlation structure for the NAA deviations. Must be a single character string described below or a vector of length `n_stocks`. If length = 1, assumptions will be applied to all stocks. 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).

\$cor_vals Array (`n_stocks` x `n_regions` x 3) of initial correlation values to use for the NAA deviations. The first value correspond to correlation across age. The second is for yearly correlation for NAA deviations for all ages if recruitment is not decoupled, or otherwise just ages after recruitment. The third correlation is for annual recruitment deviations and used only when recruitment is decoupled. If unspecified all initial values are 0. If `NAA_re$cor = "iid"`, values are not used. If `NAA_re$cor = "ar1_a"`, those for yearly correlation are not used, and vice versa for `"ar1_y"`.

\$cor_map `n_stocks` x `n_region` matrix x 3 array of NA or integers indicating which random effects correlation parameters are estimated and whether to set any to be estimated identically. The 3 values for a given stock and region are for age, year ("survival"), and year ("recruitment"). Both age and years values are used for `NAA_re$cor = "2dar1"`, but only the appropriate values are used for `NAA_re$cor = "ar1_a"`, or `"ar1_y"`. When `NAA_re$decouple_recruitment = TRUE`, the third value is used for both `NAA_re$sigma = "rec"` or `"rec+1"`. For `NAA_re$decouple_recruitment = FALSE`, only the second value is used and applies to both recruitment and "survival" when `NAA_re$sigma = "rec+1"`, and just recruitment when `NAA_re$sigma = "rec"`. If not supplied stock-specific values for age and/or year will be estimated for all regions where `NAA_re$cor` is other than "none", "iid".

\$N1_model Character vector (n_stocks) determining which way to model the initial numbers at age:

"age-specific-fe" (default) age- and region-specific fixed effects parameters

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

"iid-re" (default) age- and region-specific iid random effects parameters. 2 parameters: mean and sd for log NAA

"ar1-re" (default) age- and region-specific random effects parameters. 3 parameters: mean and sd, and cor for log NAA

\$N1_pars An (n_stocks x n_regions x n_ages) array. If N1_model = 0, then this should be filled with the initial values to use for abundance at age by stock and region in the first year. If N1_model = 1 (equilibrium assumption), only the first two values in the ages dimension are used: the (s,r,1) value is recruitment for stock (and region) and (s,r,2) is the fully-selected equilibrium fishing mortality rate generating the rest of the numbers at age in the first year.

\$recruit_model Integer vector (n_stocks) 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 annual recruitments as fixed effects.

2 estimating a mean recruitment with annual recruitments as random effects

3 Beverton-Holt stock-recruitment with annual recruitments as random effects

4 Ricker stock-recruitment with annual recruitments as random effects

\$recruit_pars list (length = n_stocks) of vectors of initial parameters for recruitment model. If \$recruit_model is 3 or 4, parameters are "alpha" and "beta".

Value

a named list with same elements as the input provided with abundance modeling options modified.

See Also

[prepare_wham_input](#)

Examples

```
## Not run:
wham.dir <- find.package("wham")
path_to_examples <- system.file("extdata", package="wham")
asap3 <- read_asap3_dat(file.path(path_to_examples,"ex1_SNEMAYT.dat"))
input <- prepare_wham_input(asap3)
NAA <- list(sigma = "rec")
input <- set_q(input, NAA_re = NAA) #estimate recruitment as random effects
```

```
## End(Not run)
```

set_osa_obs	<i>Set up observation vector that is used by the model for likelihood calculations and one-step-ahead residuals.</i>
-------------	--

Description

Set up observation vector that is used by the model for likelihood calculations and one-step-ahead residuals.

Usage

```
set_osa_obs(input)
```

Arguments

input list containing data, parameters, map, and random elements (output from [wham::prepare_wham_input](#))

Value

the same input list as provided, but with \$obs and \$obsvec configured. This is run after any changes have been made to the data

set_q	<i>Specify model and parameter configuration for catchability</i>
-------	---

Description

Specify model and parameter configuration for catchability

Usage

```
set_q(input, catchability = NULL)
```

Arguments

input list containing data, parameters, map, and random elements (output from [prepare_wham_input](#))

catchability (optional) list specifying options for numbers-at-age random effects, initial parameter values, and recruitment model (see details)

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.

\$sigma_val Vector of initial standard deviation values to use for annual random effects for each index. Values are not used if q\$re = "none". Otherwise, a single value for all indices.

\$sigma_map Specify which sigma parameters to fix for the random effect deviations. Must be a vector with length = number of indices. Use NA to fix a parameter and integers to estimate. Use the same integer for multiple indices to share the same sigma parameter. Not used if re = 'none' for all indices.

\$cor_vals Vector of initial correlation values to use for annual random effects for each index. If unspecified all initial values are 0. Only used if q\$re = "ar1"

\$cor_map Specify which ar1 correlation parameters to fix for the random effect deviations. Must be a vector with length = number of indices. If re = 'ar1', each element (index) must be a single value. Use NA to fix a parameter and integers to estimate. Use the same integer for multiple indices to share the same correlation parameter. Not used if re = 'none' or re = 'iid' for all indices.

Value

a named list with same elements as the input provided with catchability options modified.

See Also

[prepare_wham_input](#)

Examples

```
## Not run:
wham.dir <- find.package("wham")
path_to_examples <- system.file("extdata", package="wham")
asap3 <- read_asap3_dat(file.path(path_to_examples,"ex1_SNEMAYT.dat"))
input <- prepare_wham_input(asap3)
catchability <- list(re = c("iid", "none"))
input <- set_q(input, catchability = catchability) #independent time-varying random effects on q for first survey.

## End(Not run)
```

set_selectivity	<i>Specify model and parameter configuration for selectivity</i>
-----------------	--

Description

Specify model and parameter configuration for selectivity

Usage

```
set_selectivity(input, selectivity)
```

Arguments

input	list containing data, parameters, map, and random elements (output from prepare_wham_input)
selectivity	(optional) list specifying options for selectivity blocks, models, initial parameter values, parameter fixing/mapping, and random effects (see details) set_selectivity specifies options for selectivity and allows you to overwrite existing options in the input list or as specified in the ASAP data file. If selectivity = NULL, selectivity options from input are used. prepare_wham_input(..., selectivity=selectivity) calls set_selectivity(..., selectivity=selectivity). If you already have created input with prepare_wham_input, you can also use set_selectivity(input, selectivity=selectivity) to modify the selectivity specification. selectivity is a list with the following entries: \$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. \$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 (a50 and 1/slope) or `c(0.5, 0.5, 0.5, 1, 1, 0.5)` for age-specific parameters when there are 6 ages. Default is to set at middle of parameter range. This is 0.5 for age-specific and `n.ages/2` or logistic, double-logistic, and decreasing-logistic.

\$fix_pars Alternative to `$map_pars` for specifying which selectivity parameters (only fixed effects) to fix at initial values. List of length = number of selectivity blocks. E.g. model with 3 age-specific blocks and 6 ages, `list(4:5, 4, 2:4)` will fix ages 4 and 5 in block 1, age 4 in block 2, and ages 2, 3, and 4 in block 3. Use NULL to not fix any parameters for a block, e.g. `list(NULL, 4, 2)` does not fix any pars in block 1.

\$spar_min The lower bound for selectivity parameters and is used to populate `data$selpars_lower`. List of length = number of selectivity blocks, where each item is a vector of length = number of selectivity parameters (age-specific: `n.ages`, logistic: 2, double-logistic: 4).

\$spar_max The upper bound for selectivity parameters and is used to populate `data$selpars_upper`. List of length = number of selectivity blocks, where each item is a vector of length = number of selectivity parameters (age-specific: `n.ages`, logistic: 2, double-logistic: 4).

\$map_pars Alternative to `$fix_pars` for specifying how to fix selectivity parameters (only fixed effects), corresponds to `map$logit_selpars`. List of length = number of selectivity blocks, where each item is a vector of length = number of selectivity parameters (age-specific: `n.ages`, logistic: 2, double-logistic: 4). Use NA to fix a parameter and integers to estimate. Use the same integer for multiple ages or fleets/indices to estimate a shared parameter. E.g. for a model with 3 age-specific blocks (1 fleet, 2 indices) and 6 ages, `$map_pars = list(c(1, 2, 3, NA, NA, 4), c(5, 6, 7, NA, 8, 8), c(1, 2, 3, NA, NA, 4))` will estimate ages 1-3 and 6 in block 1 (fleet), ages 1-3 and 4-5 (shared) in block 2 (index 1), and then set the index 2 (block 3) selectivity equal to the fleet.

\$sigma_vals Initial standard deviation values to use for the random effect deviations. Must be a vector with length = number of blocks. Use natural (not log) scale, must be positive. `par$sel_repars[, 1]` will be estimated on log-scale. Not used if `re = 'none'` for all blocks.

\$map_sigma Specify which SD parameters to fix for the random effect deviations. Must be a vector with length = number of blocks. Use NA to fix a parameter and integers to estimate. Use the same integer for multiple blocks to estimate a shared SD parameter. Not used if `re = 'none'` for all blocks.

\$cor_vals Initial correlation values to use for the random effect deviations. Must be a `n_selblocks x 2` integer matrix. Columns correspond to corre-

lation by age and year, respectively. If `re = 'ar1'` or `re = 'ar1_y'` only the corresponding values are used. Values must be between -1 and 1, but parameters are estimated on a logit transformed scale internally. Not used if `re = 'none'` or `re = 'iid'` for all blocks.

\$map_cor Specify which correlation parameters to fix for the random effect deviations. Must be a `n_selblocks x 2` matrix. Columns correspond to correlation by age and year, respectively. Parameters can be shared by setting corresponding values of `map_cor` to the same integer. Use NA to fix a parameter. If `re = 'ar1'` or `re = 'ar1_y'`, only the column for the corresponding correlation are used. Not used if `re = 'none'` or `re = 'iid'` for all blocks.

\$map_re Specify whether and how to estimate selectivity random effects. Structure is the same as `$map_pars`: list of length = number of selectivity blocks, where each item is a vector of length = number of selectivity parameters (age-specific: `n_ages`, logistic: 2, double-logistic: 4). Use NA to not estimate random effects for a corresponding mean parameter and integers to estimate. Use the same integer for multiple ages to estimate shared parameters within a selectivity block. For `re = "ar1_y"` or `"2dar1"`, the annual RE are unique, but identical for parameters specified to be equal. E.g. for selectivity block 1 with age-specific selectivity and annual random effect (and `n_ages = 6`), `$map_pars[[1]] = c(1, 2, 2, NA, NA, 4)` will estimate 2 sets of unique annual RE for ages 1, 6, and another unique set applied to both ages 2 and 3. No RE will be estimated for any mean parameters that are at bounds (`par_min` or `par_max`). Lack of convergence should be expected when there is only 1 unique set of annual RE and correlation across parameters is estimated (i.e., `re = "2dar1"`) or when the ratio of unique RE and mean parameters is low without annual RE (i.e., `re = "ar1"`).

Value

a named list with same elements as the input provided with selectivity options modified.

See Also

[prepare_wham_input](#)

Examples

```
## Not run:
wham.dir <- find.package("wham")
path_to_examples <- system.file("extdata", package="wham")
asap3 <- read_asap3_dat(file.path(path_to_examples, "ex1_SNEMAYT.dat"))
input <- prepare_wham_input(asap3, NAA_re = list(sigma = "rec"))
sel <- list(model=rep("logistic", input$data$n_selblocks),
  initial_pars=rep(list(c(3,3)), input$data$n_selblocks),
  fix_pars=rep(list(NULL), input$data$n_selblocks))
input <- set_selectivity(input, selectivity = sel) #logistic selectivity for all selectivity blocks

## 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()'`

Index

aic, 3

check_convergence, 4

check_estimability, 5, 10, 12

compare_wham_models, 5, 35, 36

do_reference_points, 8

do_retro_peels, 8

do_sdreport, 9

extract_fixed, 10

fit_peel, 9, 11, 37, 38

fit_tmb, 4, 5, 9, 11, 12, 14–16, 31, 33, 37, 38

fit_wham, 4, 7–12, 13, 13, 17–20, 23–25, 28, 33, 34, 36–38, 40

jitter_wham, 16

make_osa_residuals, 14, 17

MakeADFun, 13, 14

mohns_rho, 18

plot_wham_output, 19, 61

prepare_projection, 20

prepare_wham_input, 11, 23, 24, 24, 33, 34, 37, 41–43, 46, 47, 51, 53, 55–58, 60

project_wham, 8, 10, 14, 16, 20, 24, 28

read_asap3_dat, 24, 28, 33

read_asap3_fit, 6, 7, 34, 36

read_wham_fit, 7, 35, 36

reduce_input, 37

retro, 8, 9, 11, 13, 16, 19, 37

sdreport, 8, 14, 31

self_test, 38

set_age_comp, 25, 40

set_catch, 25, 26, 42

set_ecov, 24, 25, 43

set_F, 25, 46

set_indices, 25, 26, 47

set_L, 25, 48

set_M, 24, 25, 49

set_move, 25, 51

set_NAA, 25, 53

set_osa_obs, 56

set_q, 25, 56

set_selectivity, 24, 25, 58

stats::nlminb, 4, 12, 13

stats::optim, 12

summary.sdreport, 8–10, 15, 31, 38

summary.sdreport), 12

TMB::MakeADFun, 9, 11–14, 23, 27, 28, 31, 38

TMB::oneStepPredict, 14–18

TMB::sdreport, 4, 8–10, 12, 15, 31, 38

wham::prepare_wham_input, 40, 43, 46, 48, 49, 52, 56

wham_html, 20, 61