Package ‘ecostate’

July 22, 2025

Type Package

Title State-Space Mass-Balance Model for Marine Ecosystems

Version 0.2.0

Date 2024-11-15

Maintainer James T. Thorson <James.Thorson@noaa.gov>

Description Fits a state-space mass-balance model for marine ecosystems, which implements dynamics derived from 'Ecopath with Ecosim' <https://ecopath.org/> while fitting to time-series of fishery catch, biomass indices, age-composition samples, and weight-at-age data. Package 'ecostate' fits biological parameters (e.g., equilibrium mass) and measurement parameters (e.g., catchability coefficients) jointly with residual variation in process errors, and can include Bayesian priors for parameters.

License GPL-3

Depends R (>= 4.1.0), RTMB (>= 1.5.0),

Imports TMB, MASS, checkmate, ggplot2, ggnetwork, igraph

Suggests knitr, rmarkdown

RoxygenNote 7.3.2

VignetteBuilder knitr

LazyData true

URL https://james-thorson-noaa.github.io/ecostate/

BugReports https://github.com/James-Thorson-NOAA/ecostate/issues

NeedsCompilation no

Author James T. Thorson [aut, cre] (ORCID: <https://orcid.org/0000-0001-7415-1010>)

Repository CRAN

Date/Publication 2024-11-25 11:50:14 UTC

Contents

`abm3pc_sys` *Adams-Bashford-Moulton for system of equations*

Description

Third-order Adams-Bashford-Moulton predictor-corrector method.

Usage

```
abm3pc_sys(f, a, b, y0, n, Pars, ...)
```

Arguments

`f`	function in the differential equation $y' = f(x, y)$; defined as a function $R \times R^m \rightarrow R^m$, where m is the number of equations.
`a`	starting time for the interval to integrate
`b`	ending time for the interval to integrate.
`y0`	starting values at time `a`
`n`	number of steps
`Pars`	named list of parameters passed to f
`...`	additional inputs to function `f`

Details

Combined Adams-Bashford and Adams-Moulton (or: multi-step) method of third order with corrections according to the predictor-corrector approach. Copied from pracma under GPL-3, with small modifications to work with RTMB

Value

List with components x for grid points between a and b and y an n-by-m matrix with solutions for variables in columns, i.e. each row contains one time stamp.

---

add_equilibrium *Compute equilibrium values*

Description

Compute equilibrium values

Usage

```
add_equilibrium(ecoparams, scale_solver, noB_i, type_i)
```

Arguments

ecoparams	list of parameters
scale_solver	Whether to solve for ecotrophic efficiency EE given biomass B (scale_solver="simple") or solve for a combination of EE and B values
noB_i	Boolean vector indicating which taxa have no B value
type_i	character vector indicating whether a taxon is "hetero", "auto", or "detritus"

Details

Replaces NA values in ecotrophic efficiency and/or biomass with equilibrium solution, and then calculates equilibrium consumption, natural mortality, and other rates.

Value

the list of parameters with missing values in `ecoparams$B_i` and/or `ecoparams$EE_i` filled in, as well as additional values `Qe_ij`, `m0_i`, and `GE_i` compute_nll *Compute negative log-likelihood for EcoState model*

Description

Compute negative log-likelihood for EcoState model

Usage

```
compute_nll(
  p,
  taxa,
  years,
  noB_i,
  type_i,
  n_species,
  project_vars,
  DC_ij,
  Bobs_ti,
  Cobs_ti,
  Nobs_ta_g2,
  Wobs_ta_g2,
  log_prior,
  fit_eps,
  fit_nu,
  stanza_data,
  settings,
  control
)
```

Arguments

p	list of parameters
taxa	Character vector of taxa included in model.
years	Integer-vector of years included in model
noB_i	Boolean vector indicating which taxa have no B value
type_i	character vector indicating whether a taxon is "hetero", "auto", or "detritus"
n_species	number of species
project_vars	function to integrate differential equation
DC_ij	Diet projections matrix
Bobs_ti	formatted matrix of biomass data
Cobs_ti	formatted matrix of catch data
Nobs_ta_g2	formatted list of age-comp data
Wobs_ta_g2	formatted list of weight-at-age data

log_prior	A user-provided function that takes as input the list of parameters outobjenv$parList() where out is the output from ecostate(), and returns a numeric vector where the sum is the log-prior probability. For example log_prior = function(p) dnorm(p$logq_i[1], mean=0, sd=0.1, log=TRUE) specifies a lognormal prior probability for the catchability coefficient for the first taxa with logmean of zero and logsd of 0.1
fit_eps	Character-vector listing taxa for which the model should estimate annual process errors in dB/dt
fit_nu	Character-vector listing taxa for which the model should estimate annual process errors in consumption Q_ij
stanza_data	output from make_stanza_data
settings	Output from stanza_settings(), used to define age-structured dynamics (called stanza-groups).
control	output from ecostate_control

Details

Given a list of parameters, calculates the joint negative log-likelihood, where the Laplace approximation is then used to marginalize across random effects to calculate the log-marginal likelihood of fixed effects. The joint likelihood includes the fit to fishery catches, biomass indices, age-composition data, weight-at-age data, priors, and the distribution for random effects.

Value

The joint negative log-likelihood including contribution of priors and fit to data.

---

compute_tracer *Calculate tracers, e.g., trophic level*

Description

Calculate how a tracer propagates through consumption.

Usage

```
compute_tracer(
  Q_ij,
  inverse_method = c("Penrose_moore", "Standard"),
  type_i,
  tracer_i = rep(1, nrow(Q_ij))
)
```

Arguments

Q_ij	Consumption of each prey i by predator j in units biomass.
inverse_method	whether to use pseudoinverse or standard inverse
type_i	character vector indicating whether a taxon is "hetero", "auto", or "detritus"
tracer_i	an indicator matrix specifying the traver value

Details

Trophic level y_i for each predator i is defined as:

$$\mathbf{y} = \mathbf{l}\mathbf{Q}^* + \mathbf{1}$$

where $\mathbf{Q}*$ is the proportion consumption for each predator (column) of different prey (rows). We identify primary producers as any taxa with no consumption (a column of 0s), and assign them as the first trophic level.

More generically, a tracer might be used to track movement of biomass through consumption. For example, if we have a tracer x_i that is 1 for the base of the pelagic food chain, and 0 otherwise, then we can calculate the proportion of pelagic vs. nonpelagic biomass for each taxon:

$$\mathbf{y} = \mathbf{l}\mathbf{Q}^* + \mathbf{x}$$

This then allows us to separate alternative components of the foodweb.

Value

The vector

$$\mathbf{y_i}$$

resulting from tracer

$$\mathbf{x_i}$$

`dBdt` *Dynamics from EcoSim*

Description

Compute system of differential equations representing EcoState dynamics derived from EcoSim.

Usage

```
dBdt(
  Time,
  State,
  Pars,
  type_i,
  n_species,
  F_type = "integrated",
  what = "dBdt"
)
```

Arguments

`Time`	not used
`State`	vector of state variables to integrate
`Pars`	list of parameters governing the ODE
`type_i`	type for each taxon
`n_species`	number of species
`F_type`	whether to integrate catches along with biomass (`"integrated"`) or calculate catches from the Baranov catch equation applied to average biomass (`"averaged"`)
`what`	what output to produce

Details

This function has syntax designed to match `pracma` solvers.

Value

An object (list) of ranges. Elements include:

G_i Biomass growth per time

g_i Biomass growth per time per biomass

M2_i Consumptive mortality per time

m2_i Consumptive mortality per time per biomass

M_i Natural mortality per time

m_i Natural mortality per time per biomass (i.e., m2_i + m0_i)

Q_ij Consumption per time for prey (rows) by predator (columns)

`ddirmult` *Dirichlet-multinomial*

Description

Allows data-weighting as parameter

Usage

```
ddirmult(x, prob, ln_theta, log = TRUE)
```

Arguments

`x`	numeric vector of observations across categories
`prob`	numeric vector of category probabilities
`ln_theta`	logit-ratio of effective and input sample size
`log`	whether to return the log-probability or not

Value

The log-likelihood resulting from the Dirichlet-multinomial distribution

Examples

```
library(RTMB)
prob = rep(0.1,10)
x = rmultinom( n=1, prob=prob, size=20 )[,1]
f = function( ln_theta ) ddirmult(x, prob, ln_theta)
f( 0 )
F = MakeTape(f, 0)
F$jacfun()(0)
```

---

eastern_bering_sea *eastern Bering Sea ecosystem data*

Description

Data used to demonstrate a Model of Intermediate Complexity (MICE) for the eastern Bering Sea. `data(eastern_bering_sea)` loads a list that includes four components:

- `Survey` is a long-form data-frame with three columns, providing the Year, Mass (in relative units for most taxa, and million metric tons for Pollock, Cod, Arrowtooth, and NFS), and Taxon for each year with available data
- `Catch` is a long-form data-frame with three columns, providing the Year, Mass (in million metric tons), and Taxon for each year with available data
- `P_over_B` is a numeric vector with the unitless ratio of biomass production to population biomass for each taxon
- `Q_over_B` is a numeric vector with the unitless ratio of biomass consumption to population biomass for each taxon
- `Diet_proportions` is a numeric matrix where each column lists the proportion of biomass consumed that is provided by each prey (row)

Usage

```
data(eastern_bering_sea)
```

Details

The data compiled come from a variety of sources:

- Northern fur seal (NFS) survey is an absolute index, corrected for proportion of time spent in the eastern Bering Sea. NFS QB is developed from a bioenergetic model and also corrected for seasonal residency. Both are provided by Elizabeth McHuron. It is post-processed in a variety of ways, and not to be treated as an index of abundance for NFS for other uses.

- Pollock, cod, and arrowtooth surveys are from a bottom trawl survey, and cod and arrowtooth are treated as an absolute index.
- Copepod and other zooplankton are from an oblique tow bongo net survey, with data provided by Dave Kimmel. It is then post-processed to account for spatially and seaonally imbalanced data.
- Other P_over_B, Q_over_B and Diet_proportions values are derived from Rpath models, provided by Andy Whitehouse.
- Primary producers is an annual index of relative biomass, developed from monthly satellite measurements and provided by Jens Nielsen. See Thorson et al. (In review) for more details regarding data standardization and sources

---

ecostate *fit EcoState model*

Description

Estimate parameters for an EcoState model

Usage

```
ecostate(
  taxa,
  years,
  catch = data.frame(Year = numeric(0), Mass = numeric(0), Taxon = numeric(0)),
  biomass = data.frame(Year = numeric(0), Mass = numeric(0), Taxon = numeric(0)),
  agecomp = list(),
  weight = list(),
  PB,
  QB,
  B,
  DC,
  EE,
  X,
  type,
  U,
  fit_B = vector(),
  fit_Q = vector(),
  fit_B0 = vector(),
  fit_EE = vector(),
  fit_PB = vector(),
  fit_QB = vector(),
  fit_eps = vector(),
  fit_nu = vector(),
  log_prior = function(p) 0,
  settings = stanza_settings(taxa = taxa),
  control = ecostate_control()
)
```

Arguments

- `taxa` — Character vector of taxa included in model.
- `years` — Integer-vector of years included in model
- `catch` — long-form data frame with columns `Mass`, `Year` and `Taxon`
- `biomass` — long-form data frame with columns `Mass`, `Year` and `Taxon`, where `Mass` is assumed to have the same units as `catch`
- `agecomp` — a named list, with names corresponding to `stanza_groups`, where each list-element is a matrix with rownames for `years` and colnames for integer ages, where NA excludes the entry from inclusion and the model computes the likelihood across included ages in a given year, and the rowsum is the input-sample size for a given year
- `weight` — a named list, with names corresponding to `stanza_groups`, where each list-element is a matrix with rownames for `years` and colnames for integer ages, where NA excludes the entry from inclusion and the model computes the log-normal likelihood for weight-at-age in each specified age-year combination
- `PB` — numeric-vector with names matching `taxa`, providing the ratio of production to biomass for each taxon
- `QB` — numeric-vector with names matching `taxa`, providing the ratio of consumption to biomass for each taxon
- `B` — numeric-vector with names matching `taxa`, providing the starting (or fixed) value for equilibrium biomass for each taxon
- `DC` — numeric-matrix with rownames and colnames matching `taxa`, where each column provides the diet proportion for a given predator
- `EE` — numeric-vector with names matching `taxa`, providing the proportion of proportion of production that is subsequently modeled (termed ecotrophic efficiency)
- `X` — numeric-matrix with rownames and colnames matching `taxa`, where each element gives the vulnerability parameter for a given interaction.
- `type` — character-vector with names matching `taxa` and elements `c("auto","hetero","detritus")`, indicating whether each taxon is a primary producer, consumer/predator, or detritus, respectively.
- `U` — numeric-vector with names matching `taxa`, providing the proportion of consumption that is unassimilated and therefore exported to detritus
- `fit_B` — Character-vector listing `taxa` for which equilibrium biomass is estimated as a fixed effect
- `fit_Q` — Character-vector listing `taxa` for which the catchability coefficient is estimated as a fixed effect
- `fit_B0` — Character-vector listing `taxa` for which the ratio of initial to equilibrium biomass is estimated as a fixed effect
- `fit_EE` — Character-vector listing `taxa` for which ecotrophic efficiency is estimated.
- `fit_PB` — Character-vector listing `taxa` for which equilibrium production per biomass is estimated. Note that it is likely a good idea to include a prior for any species for which this is estimated.

`fit_QB` Character-vector listing `taxa` for which equilibrium consumption per biomass is estimated. Note that it is likely a good idea to include a prior for any species for which this is estimated.

`fit_eps` Character-vector listing `taxa` for which the model should estimate annual process errors in dB/dt

`fit_nu` Character-vector listing `taxa` for which the model should estimate annual process errors in consumption `Q_ij`

`log_prior` A user-provided function that takes as input the list of parameters `out$obj$env$parList()` where `out` is the output from `ecostate()`, and returns a numeric vector where the sum is the log-prior probability. For example `log_prior = function(p) dnorm( p$logq_i[1], mean=0, sd=0.1, log=TRUE)` specifies a lognormal prior probability for the catchability coefficient for the first `taxa` with logmean of zero and logsd of 0.1

`settings` Output from `stanza_settings()`, used to define age-structured dynamics (called stanza-groups).

`control` Output from `ecostate_control()`, used to define user settings.

Details

All `taxa` must be included in `QB`, `PB`, `B`, and `DC`, but additional taxa can be in `QB`, `PB`, `B`, and `DC` that are not in `taxa`. So `taxa` can be used to redefine the set of modeled species without changing other inputs

Value

An object (list) of S3-class `ecostate`. Elements include:

obj RTMB object from `MakeADFun`

tmb_inputs The list of inputs passed to `MakeADFun`

opt The output from `nlminb`

sdrep The output from `sdreport`

interal Objects useful for package function, i.e., all arguments passed during the call

rep report file, including matrix `B_ti` for biomass in each year `t` and taxon `i`, `g_ti` for growth rate per biomass, and see `dBdt` for other quantities reported by year

derived derived quantity estimates and standard errors, for `rep` objects as requested

call function call record

run_time Total runtime

This S3 class then has functions `summary`, `print`, and `logLik`

References

Introducing the state-space mass-balance model:

Thorson, J. Kristensen, K., Aydin, K., Gaichas, S., Kimmel, D.G., McHuron, E.A., Nielsen, J.N., Townsend, H., Whitehouse, G.A (In press). The benefits of hierarchical ecosystem models: demonstration using a new state-space mass-balance model EcoState. Fish and Fisheries.

ecostate_control *Detailed control for ecostate structure*

Description

Define a list of control parameters.

Usage

```
ecostate_control(
  nlminb_loops = 1,
  newton_loops = 0,
  eval.max = 1000,
  iter.max = 1000,
  getsd = TRUE,
  silent = getOption("ecostate.silent", TRUE),
  trace = getOption("ecostate.trace", 0),
  verbose = getOption("ecostate.verbose", FALSE),
  profile = c("logF_ti", "log_winf_z", "s50_z", "srate_z"),
  random = c("epsilon_ti", "alpha_ti", "nu_ti", "phi_tg2"),
  tmb_par = NULL,
  map = NULL,
  getJointPrecision = FALSE,
  integration_method = c("ABM", "RK4", "ode23", "rk4", "lsoda"),
  process_error = c("epsilon", "alpha"),
  n_steps = 10,
  F_type = c("integrated", "averaged"),
  derived_quantities = c("h_g2", "B_ti", "B0_i"),
  scale_solver = c("joint", "simple"),
  inverse_method = c("Standard", "Penrose_moore"),
  tmbad.sparse_hessian_compress = 1,
  start_tau = 0.001
)
```

Arguments

`nlminb_loops`	Integer number of times to call `stats::nlminb()`.
`newton_loops`	Integer number of Newton steps to do after running `stats::nlminb()`.
`eval.max`	Maximum number of evaluations of the objective function allowed. Passed to `control` in `stats::nlminb()`.
`iter.max`	Maximum number of iterations allowed. Passed to `control` in `stats::nlminb()`.
`getsd`	Boolean indicating whether to call `TMB::sdreport()`
`silent`	Disable terminal output for inner optimizer?
`trace`	Parameter values are printed every `trace` iteration for the outer optimizer. Passed to `control` in `stats::nlminb()`.

verbose	Output additional messages about model steps during fitting?
profile	parameters that are profiled across, passed to MakeADFun
random	parameters that are treated as random effects, passed to MakeADFun
tmb_par	list of parameters for starting values, with shape identical to tinyVAST(...)$internal$parlist
map	list of mapping values, passed to RTMB::MakeADFun
getJointPrecision	whether to get the joint precision matrix. Passed to sdreport.
integration_method	What numerical integration method to use. "ABM" uses a native-R versions of Adam-Bashford, "RK4" uses a native-R version of Runge-Kutta-4, and "ode23" uses a native-R version of adaptive Runge-Kutta-23, where all are adapted from pracma functions. "rk4" and lsoda use those methods from deSolve::ode as implemented by RTMBode::ode
process_error	Whether to include process error as a continuous rate (i.e., an "innovation" parameterization, process_error="epsilon") or as a discrete difference between expected and predicted biomass (i.e., a "state-space" parameterization), process_error="alpha"The former is more interpretable, whereas the latter is much more computationally efficient.
n_steps	number of steps used in the ODE solver for biomass dynamics
F_type	whether to integrate catches along with biomass ("integrated") or calculate catches from the Baranov catch equation applied to average biomass ("averaged")
derived_quantities	character-vector listing objects to ADREPORT
scale_solver	Whether to solve for ecotrophic efficiency EE given biomass B (scale_solver="simple") or solve for a combination of EE and B values
inverse_method	whether to use pseudoinverse or standard inverse
tmbad.sparse_hessian_compress	passed to TMB::config(), and enabling an experimental feature to save memory when first computing the inner Hessian matrix. Using tmbad.sparse_hessian_compress=1 seems to have no effect on the MLE (although users should probably confirm this), and hugely reduces memory use in both small and large models. Using tmbad.sparse_hessian_compress=1 seems to hugely speed up the model-fitting with a large model but results in a small decrease in speed for model-fitting with a small model.
start_tau	Starting value for the standard deviation of process errors

Value

An S3 object of class "ecostate_control" that specifies detailed model settings, allowing user specification while also specifying default values

ginv *Penrose-Moore pseudoinverse*

Description

Extend `MASS:ginv` to work with RTMB

Usage

```
ginv(x)
```

Arguments

`x`	Matrix used to compute pseudoinverse

logLik.ecostate *Marginal log-likelihood*

Description

Extract the (marginal) log-likelihood of a ecostate model

Usage

```
## S3 method for class 'ecostate'
logLik(object, ...)
```

Arguments

`object`	Output from `ecostate`
`...`	Not used

Value

object of class `logLik` with attributes

`val`	log-likelihood
`df`	number of parameters

Returns an object of class logLik. This has attributes "df" (degrees of freedom) giving the number of (estimated) fixed effects in the model, abd "val" (value) giving the marginal log-likelihood. This class then allows `AIC` to work as expected.

ode23 — *Non-stiff (and stiff) ODE solvers*

Description

Runge-Kutta (2, 3)-method with variable step size, resp

Usage

```
ode23(f, a, b, y0, n, Pars, rtol = 0.001, atol = 1e-06)
```

Arguments

`f`	function in the differential equation $y' = f(x, y)$; defined as a function $R \times R^m \to R^m$, where m is the number of equations.
`a`	starting time for the interval to integrate
`b`	ending time for the interval to integrate.
`y0`	starting values at time `a`
`n`	Not used
`Pars`	named list of parameters passed to f
`rtol`	relative tolerance.
`atol`	absolute tolerance.

Details

Copied from pracma under GPL-3, with small modifications to work with RTMB. This can be used to simulate dynamics, but not during estimation

Value

List with components t for time points between a and b and y an n-by-m matrix with solutions for variables in columns, i.e. each row contains one time stamp.

plot_foodweb — *Plot foodweb*

Description

Plot consumption as a directed graph including all taxa (vertices) and biomass consumed (arrows). Taxa are located using tracers, where by default the y-axis is trophic level. #'

Usage

```
plot_foodweb(
  Q_ij,
  type_i,
  xtracer_i,
  ytracer_i = rep(1, nrow(Q_ij)),
  B_i = rep(1, nrow(Q_ij)),
  taxa_labels = letters[1:nrow(Q_ij)],
  xloc,
  yloc
)
```

Arguments

`Q_ij`	Consumption of each prey i by predator j in units biomass.
`type_i`	character vector indicating whether a taxon is "hetero", "auto", or "detritus"
`xtracer_i`	tracer to use when computing x-axis values
`ytracer_i`	tracer to use when computing y-axis values
`B_i`	biomass to use when weighting taxa in plot
`taxa_labels`	character vector of labels to use for each taxon
`xloc`	x-axis location (overrides calculation using `xtracer_i`)
`yloc`	y-axis location (overrides calculation using `ytracer_i`)

Details

Trophic level l_i for each predator i is defined as:

$$\mathbf{l} - \mathbf{1} = \mathbf{l}\mathbf{Q}^*$$

where $\mathbf{Q}*$ is the proportion consumption for each predator (column) of different prey (rows). We identify primary producers as any taxa with no consumption (a column of 0s), and assign them as the first trophic level.

Value

invisibly return `ggplot` object for foodweb

print.ecostate *Print fitted ecostate object*

Description

Prints output from fitted ecostate model

Usage

```
## S3 method for class 'ecostate'
print(x, ...)
```

Arguments

x	Output from ecostate
...	Not used

Value

No return value, called to provide clean terminal output when calling fitted object in terminal.

print_ecopars *Print EcoSim parameters*

Description

Prints parameters defining EcoSim dynamics

Usage

```
print_ecopars(x, silent = FALSE)
```

Arguments

x	Output from ecostate
silent	whether to print to terminal

Value

invisibly returns table printed

rk4sys *Classical Runge-Kutta for system of equations*

Description

Classical Runge-Kutta of order 4.

Usage

```
rk4sys(f, a, b, y0, n, Pars, ...)
```

Arguments

`f`	function in the differential equation $y' = f(x, y)$; defined as a function $R \times R^m \rightarrow R^m$, where m is the number of equations.
`a`	starting time for the interval to integrate
`b`	ending time for the interval to integrate.
`y0`	starting values at time `a`
`n`	the number of steps from a to b.
`Pars`	named list of parameters passed to f
`...`	additional inputs to function `f`

Details

Classical Runge-Kutta of order 4 for (systems of) ordinary differential equations with fixed step size. Copied from pracma under GPL-3, with small modifications to work with RTMB

Value

List with components x for grid points between a and b and y an n-by-m matrix with solutions for variables in columns, i.e. each row contains one time stamp.

stanza_settings *Detailed control for stanza structure*

Description

Define a list of control parameters.

Usage

```
stanza_settings(
  taxa,
  stanza_groups,
  K,
  d,
  Wmat,
  Amax,
  SpawnX,
  Leading,
  fit_K = c(),
  fit_d = c(),
  fit_phi = vector(),
  Amat = NULL,
  Wmatslope,
  STEPS_PER_YEAR = 1,
  comp_weight = c("multinom", "dir", "dirmult")
)
```

Arguments

taxa	Character vector of taxa included in model.
stanza_groups	character-vector with names corresponding to taxa and elements specifying the multi-stanza group (i.e., age-structured population) for a given taxa
K	numeric-vector with names matching unique(stanza_groups), providing the von Bertalanffy growth coefficient for length
d	numeric-vector with names matching unique(stanza_groups), providing the von Bertalanffy allometric consumption-at-weight (default is 2/3)
Wmat	numeric-vector with names matching unique(stanza_groups), providing the weight-at-maturity relative to asymptotic weight
Amax	numeric-vector with names matching names(stanza_groups), providing the maximum age (in units years) for a given taxon (and the oldest taxon for a given stanza_group is treated as a plus-group)
SpawnX	numeric-vector with names matching unique(stanza_groups), providing the larval vulnerability (density dependence) parameter
Leading	Boolean vector with names matching names(stanza_groups), with TRUE for the taxon for which scale (B or EE) is specified or estimated, where this is then calculated determinstically for other taxa for a given stanza_group
fit_K	Character-vector listing stanza_groups for which K is estimated
fit_d	Character-vector listing stanza_groups for which d is estimated (note that this currently does not work)
fit_phi	Character-vector listing stanza_groups for which the model should estimate annual recruitment deviations, representing nonconsumptive variation in larval survival (e.g., oceanographic advection)

Amat	numeric-vector with names matching unique(stanza_groups), providing the integer age-at-maturity (in units years)
Wmatslope	numeric-vector with names matching unique(stanza_groups), providing the slope at 0.5 maturity for a logistic maturity-at-weight ogive
STEPS_PER_YEAR	integer number of Euler steps per year for calculating integrating individual weight-at-age
comp_weight	method used for weighting age-composition data

Value

An S3 object of class "stanza_settings" that specifies detailed model settings related to age-structured dynamics (e.g., stanzas), allowing user specification while also specifying default values

---

whitehouse_2021 *Full rpath inputs for eastern Bering Sea*

Description

All Rpath inputs from Whitehouse et al. 2021

Usage

```
data(whitehouse_2021)
```

Index