Package ‘bayesnec’

July 2, 2021

Title A Bayesian No-Effect- Concentration (NEC) Algorithm

Version 1.0.1


Depends R (>= 4.0), brms

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Biarch true

Imports loo, extraDistr, dplyr, tidyr, purrr, tidyselect, lazyeval, ggplot2, evaluate, rlang

Suggests rstan, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

URL https://open-aims.github.io/bayesnec/

BugReports https://github.com/open-aims/bayesnec/issues

Config/testthat/edition 3

NeedsCompilation no

Author Rebecca Fisher [aut, cre],
    Diego Barneche [aut],
    Gerard Ricardo [aut],
    David Fox [aut]

Maintainer Rebecca Fisher <r.fisher@aims.gov.au>

Repository CRAN

Date/Publication 2021-07-02 07:00:13 UTC
R topics documented:

bayesnec-package ............................................. 4
amend ......................................................... 4
amend.bayesmanecfit ........................................... 5
amend.default ................................................ 7
average_endpoints ............................................ 8
bayesmanecfit-class .......................................... 10
bayesnecfit-class ............................................ 10
beta_binomial2 ................................................ 11
beta_binomial2_lpmf .......................................... 12
beta_binomial2_rng ........................................... 12
bind_ecx ....................................................... 13
bind_nec ....................................................... 13
bnec .......................................................... 14
check_chains .................................................. 17
check_chains.bayesmanecfit ................................. 18
check_chains.bayesnecfit .................................... 18
check_chains.default ........................................ 19
check_data ..................................................... 20
check_models .................................................. 21
check_priors .................................................. 21
check_priors.bayesmanecfit ................................. 22
check_priors.bayesnecfit .................................... 23
check_priors.default ........................................ 23
compare_endpoints ............................................ 24
compare_fitted ............................................... 25
compare_posterior ............................................ 26
define_prior ................................................... 27
dispersion ...................................................... 28
ecx ............................................................ 29
ecx.bayesmanecfit ............................................ 30
ecx.bayesnecfit ................................................ 32
ecx.default .................................................... 33
expand_manec .................................................. 34
expand_nec ..................................................... 35
extract_simdat ................................................ 36
fit_bayesnec ................................................... 36
ggbnec ........................................................ 38
ggbnec.bayesmanecfit ....................................... 39
ggbnec.bayesnecfit .......................................... 40
ggbnec.default ................................................ 41
ggbnec_data .................................................... 42
ggbnec_data.bayesmanecfit ................................. 42
ggbnec_data.bayesnecfit .................................... 43
ggbnec_data.default ......................................... 44
is_bayesmanecfit ............................................. 45
is_bayesnecfit ............................................... 45
R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>is_manecsummary</td>
<td>45</td>
</tr>
<tr>
<td>is_necsummary</td>
<td>46</td>
</tr>
<tr>
<td>is_prebayesnecfit</td>
<td>46</td>
</tr>
<tr>
<td>log_lik_beta_binomial2</td>
<td>47</td>
</tr>
<tr>
<td>make_good_inits</td>
<td>47</td>
</tr>
<tr>
<td>make_inits</td>
<td>48</td>
</tr>
<tr>
<td>manecsummary-class</td>
<td>48</td>
</tr>
<tr>
<td>manec_example</td>
<td>49</td>
</tr>
<tr>
<td>models</td>
<td>50</td>
</tr>
<tr>
<td>nec</td>
<td>51</td>
</tr>
<tr>
<td>nec.bayesmanecfit</td>
<td>52</td>
</tr>
<tr>
<td>nec.bayesnecfit</td>
<td>53</td>
</tr>
<tr>
<td>nec.default</td>
<td>53</td>
</tr>
<tr>
<td>necsummary-class</td>
<td>54</td>
</tr>
<tr>
<td>nec_data</td>
<td>55</td>
</tr>
<tr>
<td>nsec</td>
<td>55</td>
</tr>
<tr>
<td>nsec.bayesmanecfit</td>
<td>57</td>
</tr>
<tr>
<td>nsec.bayesnecfit</td>
<td>58</td>
</tr>
<tr>
<td>nsec.default</td>
<td>59</td>
</tr>
<tr>
<td>plot.bayesmanecfit</td>
<td>60</td>
</tr>
<tr>
<td>plot.bayesnecfit</td>
<td>61</td>
</tr>
<tr>
<td>posterior_epred_beta_binomial2</td>
<td>63</td>
</tr>
<tr>
<td>posterior_predict_beta_binomial2</td>
<td>63</td>
</tr>
<tr>
<td>prebayesnecfit-class</td>
<td>64</td>
</tr>
<tr>
<td>predict.bayesmanecfit</td>
<td>64</td>
</tr>
<tr>
<td>predict.bayesnecfit</td>
<td>65</td>
</tr>
<tr>
<td>prep_raw_data</td>
<td>65</td>
</tr>
<tr>
<td>print.bayesmanecfit</td>
<td>66</td>
</tr>
<tr>
<td>print.bayesnecfit</td>
<td>66</td>
</tr>
<tr>
<td>print.manecsummary</td>
<td>67</td>
</tr>
<tr>
<td>print.necsummary</td>
<td>67</td>
</tr>
<tr>
<td>pull_out</td>
<td>68</td>
</tr>
<tr>
<td>pull_prior</td>
<td>68</td>
</tr>
<tr>
<td>rhat.bayesmanecfit</td>
<td>69</td>
</tr>
<tr>
<td>rhat.bayesnecfit</td>
<td>69</td>
</tr>
<tr>
<td>sample_priors</td>
<td>70</td>
</tr>
<tr>
<td>set_distribution</td>
<td>70</td>
</tr>
<tr>
<td>show_params</td>
<td>71</td>
</tr>
<tr>
<td>summary.bayesmanecfit</td>
<td>72</td>
</tr>
<tr>
<td>summary.bayesnecfit</td>
<td>72</td>
</tr>
<tr>
<td>validate_family</td>
<td>73</td>
</tr>
<tr>
<td>validate_priors</td>
<td>73</td>
</tr>
</tbody>
</table>

Index 74
Description

A No-Effect-Concentration estimation package that uses brms (https://github.com/paul-buerkner/brms) to fit concentration (dose)-response data using Bayesian methods for the purpose of estimating both ECX values, but more particularly NEC. Please see ?bnec for more details.

References


Usage

amend(
    object,
    drop,
    add,
    loo_controls,
    x_range = NA,
    precision = 1000,
    sig_val = 0.01,
    priors,
    pointwise
)

Arguments

object An object of class bayesmanecfit, as returned by bnec.
drop A character vector containing the names of model types you which to drop for the modified fit.
add A character vector containing the names of model types to add to the modified fit.
loo_controls A named list containing the desired arguments to be passed on to loo_model_weights. It can be used to change the default method from "pseudobma". See help documentation ?loo_model_weights from package loo.
amend.bayesmanecfit

x_range
A range of x values over which to consider extracting ECx.

precision
The length of the x vector used for posterior predictions, and over which to extract ECx values. Large values will be slower but more precise.

sig_val
Probability value to use as the lower quantile to test significance of the predicted posterior values against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECx curves.

priors
An object of class brmsprior which specifies user-desired prior distributions of model parameters. If missing, bnc will figure out a baseline prior for each parameter. It can also be specified as a named list where each name needs to correspond to the same string as "model". See details.

pointwise
A flag indicating whether to compute the full log-likelihood matrix at once or separately for each observation. The latter approach is usually considerably slower but requires much less working memory. Accordingly, if one runs into memory issues, pointwise = TRUE is the way to go, but will not work for the custom family beta_binomial2

Value
All successfully fitted bayesmanecfit model fits.

Examples
library(bayesnec)
data(manec_example)
exmp <- amend(manec_example, drop = "nec4param")

Usage
## S3 method for class 'bayesmanecfit'
amend(
  object,
  drop,
  add,
  loo_controls,
  x_range = NA,
  precision = 1000,
  sig_val = 0.01,
Arguments

- **object**: An object of class `bayesmanecfit`, as returned by `bnec`.
- **drop**: A character vector containing the names of model types you which to drop for the modified fit.
- **add**: A character vector containing the names of model types to add to the modified fit.
- **loo_controls**: A named list containing the desired arguments to be passed on to `loo_model_weights`. It can be used to change the default method from "pseudobma". See help documentation `?loo_model_weights` from package loo.
- **x_range**: A range of x values over which to consider extracting ECx.
- **precision**: The length of the x vector used for posterior predictions, and over which to extract ECx values. Large values will be slower but more precise.
- **sig_val**: Probability value to use as the lower quantile to test significance of the predicted posterior values against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECx curves.
- **priors**: An object of class `brmsprior` which specifies user-desired prior distributions of model parameters. If missing, `bnec` will figure out a baseline prior for each parameter. It can also be specified as a named list where each name needs to correspond to the same string as "model". See details.
- **pointwise**: A flag indicating whether to compute the full log-likelihood matrix at once or separately for each observation. The latter approach is usually considerably slower but requires much less working memory. Accordingly, if one runs into memory issues, pointwise = TRUE is the way to go, but will not work for the custom family beta_binomial2

Value

All successfully fitted `bayesmanecfit` model fits.

Examples

```r
library(bayesnec)
data(manec_example)
exmp <- amend(manec_example, drop = "nec4param")
```
**Description**

Modifies an existing `bayesmanecfit` object, for example, by adding or removing fitted models.

**Usage**

```r
## Default S3 method:
amend(
  object, 
  drop, 
  add, 
  loo_controls, 
  x_range = NA, 
  precision = 1000, 
  sig_val = 0.01, 
  priors, 
  pointwise
)
```

**Arguments**

- `object` - An object of class `bayesmanecfit`, as returned by `bnc`.
- `drop` - A character vector containing the names of model types you wish to drop for the modified fit.
- `add` - A character vector containing the names of model types to add to the modified fit.
- `loo_controls` - A named list containing the desired arguments to be passed on to `loo_model_weights`. It can be used to change the default method from "pseudobma". See help documentation `?loo_model_weights` from package loo.
- `x_range` - A range of x values over which to consider extracting ECx.
- `precision` - The length of the x vector used for posterior predictions, and over which to extract ECx values. Large values will be slower but more precise.
- `sig_val` - Probability value to use as the lower quantile to test significance of the predicted posterior values against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECx curves.
- `priors` - An object of class `brmsprior` which specifies user-desired prior distributions of model parameters. If missing, `bnc` will figure out a baseline prior for each parameter. It can also be specified as a named list where each name needs to correspond to the same string as "model". See details.
A flag indicating whether to compute the full log-likelihood matrix at once or separately for each observation. The latter approach is usually considerably slower but requires much less working memory. Accordingly, if one runs into memory issues, pointwise = TRUE is the way to go, but will not work for the custom family beta_binomial2

Value

All successfully fitted bayesmanecfit model fits.

Examples

library(bayesnec)
data(manec_example)
exmp <- amend(manec_example, drop = "nec4param")

average_endpoints

Description

Extracts posterior predicted endpoint values from a list of class bayesnecfit or bayesnecfit model fits and calculates a geometric mean.

Usage

average_endpoints(
  x,
  endpoint = "nec",
  ecx_val = 10,
  posterior = FALSE,
  type = "absolute",
  hormesis_def = "control",
  sig_val = 0.01,
  precision = 1000,
  x_range = NA,
  xform = NA,
  prob_vals = c(0.5, 0.025, 0.975)
)

Arguments

x A named list list of objects of class bayesnecfit or bayesnecfit returned by bnec.
endpoint The type of endpoint to use in the mean. Takes values "nec", "ecx" or "nsec".
ecx_val The desired percentage effect value. This must be a value between 1 and 99 (for type = "relative" and "absolute"), defaults to 10.
posterior  A logical value indicating if the full posterior sample of calculated ECx values should be returned instead of just the median and 95 credible intervals.

type  A character vector, taking values of "relative", "absolute" (the default) or "direct". See Details.

hormesis_def  A character vector, taking values of "max" or "control". See Details.

sig_val  Probability value to use as the lower quantile to test significance of the predicted posterior values, against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECx curves.

precision  The number of unique x values over which to find ECx - large values will make the ECx estimate more precise.

x_range  A range of x values over which to consider extracting ECx.

xform  A function to apply to the returned estimated concentration values.

prob_vals  A vector indicating the probability values over which to return the estimated ECx value. Defaults to 0.5 (median) and 0.025 and 0.975 (95 percent credible intervals).

Details

The geometric mean of values are simply the mean calculated on a log scale and back transformed through exp(), although we have added the capacity to accommodate zero values. Note that the function assumes that x has been modelled on the natural scale. Often C-R models are more stable on a log transformed or sqrt scaling. If the input bayesnecfit or bayesnecfit model fits are already based on a re-scaling of the x (concentration) axis, it is important to pass an appropriate xform argument to ensure these are back transformed before the the geometric mean calculation is applied.

Value

The geometric mean of the endpoints estimate values of the bayesnecfit or bayesnecfit model fits contained in x. See Details.

See Also

bnecc

Examples

library(brms)
library(bayesnec)
data(manec_example)
ecx4param <- pull_out(manec_example, model = "ecx4param")
nec4param <- pull_out(manec_example, model = "nec4param")
average_endpoints(list("nec" = ecx4param, "ecx" = nec4param), ecx_val = 50)
bayesmanecfit-class

Class bayesmanecfit of models fitted with the brms package

Description

Multiple models fitted with the bayesnec package are represented as a bayesmanecfit object, which contains the original brmsfit fitted objects, names of non-linear models that were fitted, model averaging WAIC stats, sample size, mean posterior NEC values, mean model averaged predictions on the data scale, model averaged residuals, full posterior distribution of predicted values, and summary statistics of NEC statistics.

Details

See methods(class = "bayesmanecfit") for an overview of available methods.

Slots

mod_fits A list of fitted model outputs of class prebayesnecfit for each of the fitted models.
success_models A character vector indicating the name of the successfully fitted models.
mod_stats A data.frame of model fit statistics.
sample_size The size of the posterior sample. Information on the priors used in the model.
w_nec_posterior The model-weighted posterior estimate of the NEC.
w_predicted_y The model-weighted predicted values for the observed data.
w_residuals Model-weighted residual values (i.e. observed - w_predicted_y).
w_pred_vals A list containing model-weighted posterior predicted values based on the supplied precision and x_range.
w_nec The summary stats (median and 95% credibility intervals) of w_nec_posterior.

See Also

bayesnec, bnet, bayesnecfit

bayesnecfit-class

Class bayesnecfit of models fitted with the brms package

Description

Models fitted with the bayesnec package are represented as a bayesnecfit object, which contain the original brmsfit fitted object, list of initialisation values used, name of non-linear model that was fitted, posterior predictions, posterior parameter estimates and a series of other statistics.

Details

See methods(class = "bayesnecfit") for an overview of available methods.
Slots

fit  The fitted Bayesian model of class \texttt{brmsfit}.
model  A \texttt{character} string indicating the name of the fitted model.
inits  A \texttt{list} containing the initialisation values for to fit the model.
pred_vals  A \texttt{list} containing a \texttt{data.frame} of summary posterior predicted values and a vector containing based on the supplied precision and \texttt{x\_range}.
top  The estimate for parameter "top" in the fitted model.
beta  The estimate for parameter "beta" in the fitted model.
nec  The estimated NEC.
bot  The estimate for parameter "bot" in the fitted model, NA if absent for the fitted model type.
d  The estimate for parameter "d" in the fitted model, NA if absent for the fitted model type.
slope  The estimate for parameter "slope" in the fitted model, NA if absent for the fitted model type.
ce50  The estimate for parameter "ec50" in the fitted model, NA if absent for the fitted model type.
dispersion  An estimate of dispersion.
predicted_y  The predicted values for the observed data.
residuals  Residual values of the observed data from the fitted model.
nec\_posterior  A full posterior estimate of the NEC.

See Also

\texttt{bayesnec, bnec, bayesmanecfit}
beta_binomial2_lpmf

Description
Beta-binomial wrapper LPMF

Usage
beta_binomial2_lpmf(y, mu, phi, trials)

Arguments
- y: vector of observation successes.
- mu: posterior mu.
- phi: posterior phi.
- trials: vector of observation trials.

Value
A numeric value or vector containing the probability density of the beta binomial distribution

beta_binomial2_rng

Description
Beta-binomial wrapper RNG

Usage
beta_binomial2_rng(mu, phi, trials)

Arguments
- mu: posterior mu.
- phi: posterior phi.
- trials: vector of observation trials.

Value
A numeric value or vector containing random values of the beta binomial distribution
**bind_ecx**

**Description**

bind_nec

**Usage**

bind_ecx(data, ecx_vals)

**Arguments**

- **data**
  - A data.frame.
- **ecx_vals**
  - A numeric vector containing the mean, and 95% credible intervals of ECx values.

**Value**

A data.frame.

---

**bind_nec**

**Description**

bind_nec

**Usage**

bind_nec(data, nec_vals)

**Arguments**

- **data**
  - A data.frame.
- **nec_vals**
  - A numeric vector containing the mean, and 95% credible intervals of NEC values.

**Value**

A data.frame.
**bnec**

---

**Description**

Fits a variety of NEC models using Bayesian analysis and provides a model averaged predictions based on WAIC model weights.

**Usage**

```r
bnec(
  data,
  x_var,
  y_var,
  model = "all",
  trials_var = NA,
  family = NULL,
  priors,
  x_range = NA,
  precision = 1000,
  sig_val = 0.01,
  iter = 10000,
  warmup = floor(iter/10) * 9,
  inits,
  pointwise,
  sample_prior = "yes",
  loo_controls = list(method = "pseudobma"),
  random = NA,
  random_vars = NA,
  ...
)
```

**Arguments**

- `data` A `data.frame` containing the data to use for the model.
- `x_var` A `character` indicating the column heading containing the concentration (x) variable.
- `y_var` A `character` indicating the column heading containing the response (y) variable.
- `model` A `character` vector indicating the model(s) to fit. See Details for more information.
- `trials_var` The column heading indicating the column for the number of "trials" for binomial or beta_binomial2 response data. If not supplied, the model may run but will not be the model you intended!
- `family` Either a `character` string, a function, or an object of class `family` defining the statistical distribution (family) to use for the y (response) data. See details.
bnec

**priors**
An object of class `brmsprior` which specifies user-desired prior distributions of model parameters. If missing, `bnec` will figure out a baseline prior for each parameter. It can also be specified as a named list where each name needs to correspond to the same string as "model". See details.

**x_range**
A range of x values over which to consider extracting ECx.

**precision**
The length of the x vector used for posterior predictions, and over which to extract ECx values. Large values will be slower but more precise.

**sig_val**
Probability value to use as the lower quantile to test significance of the predicted posterior values against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECx curves.

**iter**
The number of iterations to be passed to `brm`. Defaults to 2e3 to be consistent with brms defaults.

**warmup**
A positive integer specifying number of warmup (a.k.a. burnin) iterations. This also specifies the number of iterations used for stepsize adaptation, so warmup samples should not be used for inference. The number of warmup should not be larger than "iter" and the default is "floor(iter / 5) * 4".

**inits**
Optional. Initialisation values. Must be a list of "n" names lists, where "n" corresponds to the number of chains, and names correspond to the parameter names of a given model.

**pointwise**
A flag indicating whether to compute the full log-likelihood matrix at once or separately for each observation. The latter approach is usually considerably slower but requires much less working memory. Accordingly, if one runs into memory issues, pointwise = TRUE is the way to go, but will not work for the custom family beta_binomial2

**sample_prior**
Indicate if samples from priors should be drawn additionally to the posterior samples. Options are "no", "yes" (the default), and "only". Among others, these samples can be used to calculate Bayes factors for point hypotheses via hypothesis.

**loo_controls**
A named list containing the desired arguments to be passed on to `loo_model_weights`. It sets the default wi_method to "pseudobma". See help documentation ?loo_model_weights from package loo.

**random**
= A named list containing the random model formula to apply to model parameters.

**random_vars**
= A character vector containing the names of the columns containing the variables used in the random model formula.

... Further arguments to `brm` via `fit_bayesnec`.

**Details**
As some concentration-response data will use zero concentration which can cause numerical estimation issues, a small offset is added (1 / 10th of the next lowest value) to zero values of concentration where x_var are distributed on a continuous scale from 0 to infinity, or are bounded to 0, or 1.

The argument `family` indicates the family to use for the response variable in the `brm` call, and may currently be "Beta" / Beta / Beta(), "binomial" / binomial / binomial(), "beta_binomial2" /
beta_binomial2, "Gamma" / Gamma / Gamma(), "gaussian" / gaussian / gaussian(), "negbinomial" / negbinomial / negbinomial(), or "poisson" / poisson / poisson(). Notice that families Beta and negbinomial are exported objects of package brms, so the user needs to load brms before calling these families.

Other families can be added as required, please raise an issue on the GitHub development site if your required family is not currently available.

If not supplied, the appropriate distribution will be guessed based on the characteristics of the input data through check_data. Guesses include all of the above families but "negbinomial" and "beta_binomial2" because these requires knowledge on whether the data is over-dispersed. As explained below in the Return section, the user can extract the dispersion parameter from a bnec call, and if they so wish, can refit the model using the "negbinomial" family.

The argument model may be a character string indicating the names of the desired model. see ?models for more details, and the list of models available. If a recongised model name is provided a single model of the specified type is fit, and bnec returns a model object of class bayesnecfit.

If a vector of two or more of the available models is supplied, bnec returns a model object of class bayesmanecfit containing model averaged predictions for the supplied models, providing they were successfully fitted.

Model averaging is achieved through a weighted sample of each fitted models posterior predictions, with weights derived using the loo_model_weights from brms. Individual model fits can be extracted from the mod_fits element and can be examined individually.

model may also be one of "all", meaning all of the available models will be fit; "ecx" meaning only models excluding a specific NEC step parameter will be fit; "nec" meaning only models with a specific NEC step parameter will be fit; or "bot_free" meaning only models without a "bot" parameter (without a bottom plateau) will be fit. Notice that if one of these group strings is provided together with a user-specified named list for the argument priors, the list names need to contain the actual model names, and not the group string, e.g. if model = "ecx" and priors = my_priors then names(my_priors) must contain models("ecx"). To check available models and associated parameters for each group, use the function models or to check the parameters of a specific model use the function show_params.

Models are fitted using model formula passed to brms.

All models provide an estimate for NEC. For model types with "nec" as a prefix, NEC is directly estimated as parameter "nec" in the model. Models with "ecx" as a prefix are continuous curve models, typically used for extracting ECx values from concentration response data. In this instance the NEC value is defined as the concentration at which there is a user supplied (see sig_val) percentage certainty (based on the Bayesian posterior estimate) that the response falls below the estimated value of the upper asymptote (top) of the response (i.e. the response value is significantly lower than that expected in the case of no exposure). The default value for sig_val is 0.01, which corresponds to an alpha value of 0.01 for a one-sided test of significance.

Value

If argument model is a single string, then an object of class bayesnecfit; if many strings or a set, an object of class bayesmanecfit.

Examples
check_chains

library(brms)
library(bayesnec)
data(nec_data)

# A single model
exmp_a <- bnec(data = nec_data, x_var = "x", y_var = "y",
               model = "nec4param", chains = 2)
# Two models model
exmp_b <- bnec(data = nec_data, x_var = "x", y_var = "y",
               model = c("nec4param", "ecx4param"), chains = 2)

---

Description

Plots mcmc chains for a bayesnecfit or bayesmanecfit model fit as returned by bnec.

Usage

check_chains(x, ...)

Arguments

x An object of class bayesnecfit or bayesmanecfit as returned by bnec.
...

arguments used when class is bayesmanecfit.

Value

No return value, generates a plot or writes a pdf to file.

Examples

library(bayesnec)
data(manec_example)

# print to device
check_chains(manec_example)
Description

Plots MCMC chains for a `bayesnecfit` model fit as returned by `bnec`.

Usage

```r
## S3 method for class 'bayesnecfit'
check_chains(x, ..., filename = NA)
```

Arguments

- `x`: An object of class `bayesnecfit` or `bayesmanecfit` as returned by `bnec`.
- `...`: arguments used when class is `bayesmanecfit`.
- `filename`: An optional character vector to be used as a pdf filename in the case of a `bayesmanecfit`. Any non empty character string will indicate the user wants to save the plots.

Value

No return value, generates a plot or writes a pdf to file.

Examples

```r
library(bayesnec)
data(manec_example)

# print to device
check_chains(manec_example)
```

Description

Plots MCMC chains for a `bayesnecfit` model fit as returned by `bnec`.

Usage

```r
## S3 method for class 'bayesnecfit'
check_chains(x, ...)
```
check_chains.default

Arguments

x  An object of class bayesnecfit or bayesmanecfit as returned by b nec.  
... arguments used when class is bayesmanecfit.

Value

No return value, generates a plot or writes a pdf to file.

Examples

library(bayesnec)
data(manec_example)

# print to device
check_chains(manec_example)
check_data

Description

Check data input for a Bayesian NEC model fit

Usage

check_data(data, x_var, y_var, trials_var, family, model, random_vars = NA)

Arguments

data: A data.frame containing the data to use for the model.
x_var: A character indicating the column heading containing the concentration (x) variable.
y_var: A character indicating the column heading containing the response (y) variable.
trials_var: The column heading indicating the column for the number of "trials" for binomial or beta_binomial2 response data. If not supplied, the model may run but will not be the model you intended!
family: Either a character string, a function, or an object of class family defining the statistical distribution (family) to use for the y (response) data. See details.
model: A character vector indicating the model(s) to fit. See Details for more information.
random_vars: = A character vector containing the names of the columns containing the variables used in the random model formula.

Details

This is a wrapper function to test input data criteria and find the correct priors for use in fit_bayesnec.

Value

A list of modified elements necessary for fit_bayesnec.
check_models

Description
Check model input for a Bayesian model fit

Usage
check_models(model, family)

Arguments
model
A character vector indicating the model(s) to fit. See Details for more information.

family
Either a character string, a function, or an object of class family defining the statistical distribution (family) to use for the y (response) data. See details.

Details
This is a wrapper function to test input model criteria and find the correct models for use in fit_bayesnec.

Value
A list of modified elements necessary for fit_bayesnec.

check_priors

Description
Plots the prior and posterior parameter probability densities from an object of class bayesnecfit or bayesmanecfit.

Usage
check_priors(object, ...)

Arguments
object
An object of class bayesnecfit or bayesmanecfit returned by bnc.

... Unused.
check_priors.bayesmanecfit

### Value
A plot of the prior and posterior parameter probability densities

### See Also
bnec

### Examples
```r
library(bayesnec)
data(manec_example)
check_priors(manec_example)
```

### Description
Plots the prior and posterior parameter probability densities from an object of class `bayesmanecfit`.

### Usage
```r
## S3 method for class 'bayesmanecfit'
check_priors(object, ..., filename = NA)
```

### Arguments
- **object**: An object of class `bayesmanecfit` returned by `bnec`.
- **...**: Unused.
- **filename**: An optional character vector to be used as a pdf filename in the case of a `bayesmanecfit`. Any non empty character string will indicate the user wants to save the plots.

### Value
A plot of the prior and posterior parameter probability densities

### See Also
bnec

### Examples
```r
library(bayesnec)
data(manec_example)
check_priors(manec_example)
```
Description

Plots the prior and posterior parameter probability densities from an object of class `bayesnecfit`.

Usage

```r
## S3 method for class 'bayesnecfit'
check_priors(object, ...)
```

Arguments

- `object`: An object of class `bayesnecfit` returned by `bnec`.
- `...`: Unused.

Value

A plot of the prior and posterior parameter probability densities

See Also

`bnec`

Examples

```r
library(bayesnec)
data(manec_example)
check_priors(manec_example)
```

Description

Plots the prior and posterior parameter probability densities from an object of class `bayesnecfit` or `bayesnecfit`.

Usage

```r
## Default S3 method:
check_priors(object, ...)
```
Arguments

object  An object of class `bayesnecfit` or `bayesmanecfit` returned by `bnec`.

Value

A plot of the prior and posterior parameter probability densities

See Also

`bnec`

Examples

```r
library(bayesnec)
data(manec_example)
check_priors(manec_example)
```

Description

Extracts posterior predicted values from a list of class `bayesnecfit` or `bayesnecfit` model fits and compares these via bootstrap re sampling.

Usage

```r
compare_endpoints(
  x, 
  comparison = "nec", 
  ecx_val = 10, 
  type = "absolute", 
  hormesis_def = "control", 
  sig_val = 0.01, 
  precision, 
  x_range = NA
)
```

Arguments

- `x`: A named list list of objects of class `bayesnecfit` or `bayesnecfit` returned by `bnec`.
- `comparison`: The posterior predictions to compare, takes values of "nec", "nsec", "ecx" or "fitted".
compare_fitted

ecx_val  The desired percentage effect value. This must be a value between 1 and 99 (for type = "relative" and "absolute"), defaults to 10.

type  A character vector, taking values of "relative", "absolute" (the default) or "direct". See Details.

hormesis_def  A character vector, taking values of "max" or "control". See Details.

sig_val  Probability value to use as the lower quantile to test significance of the predicted posterior values. against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECx curves.

precision  The number of unique x values over which to find ECx - large values will make the ECx estimate more precise.

x_range  A range of x values over which to consider extracting ECx.

Value

A named list containing bootstrapped differences in posterior predictions of the bayesnecfit or bayesnecfit model fits contained in x. See Details.

See Also

bnec

Examples

library(bayesnec)
data(manec_example)
ecx4param <- pull_out(manec_example, model = "ecx4param")
nec4param <- pull_out(manec_example, model = "nec4param")
compare_endpoints(list("nec" = ecx4param, "ecx" = nec4param), ecx_val = 50)

compare_fitted

Description

Extracts posterior predicted values from a list of class bayesnecfit or bayesnecfit model fits and compares these across a vector of fitted values.

Usage

compare_fitted(x, precision = 50, x_range = NA)
Arguments

- **x**: A named list of objects of class `bayesnecfit` or `bayesnecfit` returned by `bnec`.
- **precision**: The number of unique x values over which to find ECx - large values will make the ECx estimate more precise.
- **x_range**: A range of x values over which to consider extracting ECx.

Value

A named list containing bootstrapped differences in posterior predictions of the `bayesnecfit` or `bayesnecfit` model fits contained in x. See Details.

See Also

- `bnec`

Examples

```r
library(bayesnec)
data(manec_example)
ecx4param <- pull_out(manec_example, model = "ecx4param")
nec4param <- pull_out(manec_example, model = "nec4param")
compare_fitted(list("nec" = ecx4param, "ecx" = nec4param))
```

Description

Extracts posterior predicted values from a list of class `bayesnecfit` or `bayesnecfit` model fits and compares these via bootstrap re sampling.

Usage

```r
compare_posterior(
  x,
  comparison = "nec",
  ecx_val = 10,
  type = "absolute",
  hormesis_def = "control",
  sig_val = 0.01,
  precision,
  x_range = NA
)
```
Arguments

- **x**: A named list of objects of class `bayesnecfit` or `bayesnecfit` returned by `bnec`.
- **comparison**: The posterior predictions to compare, takes values of "nec", "nsec", "ecx" or "fitted".
- **ecx_val**: The desired percentage effect value. This must be a value between 1 and 99 (for type = "relative" and "absolute"), defaults to 10.
- **type**: A character vector, taking values of "relative", "absolute" (the default) or "direct". See Details.
- **hormesis_def**: A character vector, taking values of "max" or "control". See Details.
- **sig_val**: Probability value to use as the lower quantile to test significance of the predicted posterior values. against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECX curves.
- **precision**: The number of unique x values over which to find ECX - large values will make the ECX estimate more precise.
- **x_range**: A range of x values over which to consider extracting ECX.

Value

A named list containing bootstrapped differences in posterior predictions of the `bayesnecfit` or `bayesnecfit` model fits contained in x. See Details.

See Also

- `bnec`

Examples

```r
library(bayesnec)
data(manec_example)
ecx4param <- pull_out(manec_example, model = "ecx4param")
nec4param <- pull_out(manec_example, model = "nec4param")
compare_posterior(list("nec" = ecx4param, "ecx" = nec4param), ecx_val = 50)
```

Description

Generates prior model objects to pass to `brms`

Usage

```r
define_prior(model, family, predictor, response)
```
Arguments

model A character vector indicating the model(s) to fit. See Details for more information.
family Either a character string, a function, or an object of class family defining the statistical distribution (family) to use for the y (response) data. See details.
predictor The predictor variable for the NEC model fit.
response The response variable for the NEC model fit.

Value

A list containing the prior and the string defining response distribution family to use.

---
dispersion Posterior dispersion

Description

Calculates posterior dispersion metric

Usage

dispersion(model, summary = FALSE)

Arguments

model An object of class brmsfit whose distribution family is either gaussian, poisson or binomial.
summary Logical. Should summary stats be returned instead of full vector? Defaults to FALSE.

Details

This function calculates a dispersion metric which takes the ratio between the observed relative to simulated Pearson residuals sums of squares.

Value

If summary is FALSE, an n-long numeric vector containing the dispersion metric, where n is the number of post warm-up posterior draws from the brmsfit object. If TRUE, then a data.frame containing the summary stats (mean, median, 95% highest density intervals) of the dispersion metric.
Examples

```r
library(bayesnec)
data(manec_example)
exmp_brmsfit <- manec_example$mod.fits$nec4param$fit
dispersion(exmp_brmsfit, summary = TRUE)
```

ecx  ecx

Description

Extracts the predicted ECx value as desired from an object of class `bayesnecfit` or `bayesmanecfit`.

Usage

```r
ecx(
  object,
  ecx_val = 10,
  precision = 1000,
  posterior = FALSE,
  type = "absolute",
  hormesis_def = "control",
  x_range = NA,
  xform = NA,
  prob_vals = c(0.5, 0.025, 0.975)
)
```

Arguments

- **object**: An object of class `bayesnecfit` or `bayesmanecfit` returned by `bnec`.
- **ecx_val**: The desired percentage effect value. This must be a value between 1 and 99 (for `type = "relative"` and "absolute"), defaults to 10.
- **precision**: The number of unique x values over which to find ECx - large values will make the ECx estimate more precise.
- **posterior**: A logical value indicating if the full posterior sample of calculated ECx values should be returned instead of just the median and 95 credible intervals.
- **type**: A character vector, taking values of "relative", "absolute" (the default) or "direct". See Details.
- **hormesis_def**: A character vector, taking values of "max" or "control". See Details.
- **x_range**: A range of x values over which to consider extracting ECx.
- **xform**: A function to apply to the returned estimated concentration values.
- **prob_vals**: A vector indicating the probability values over which to return the estimated ECx value. Defaults to 0.5 (median) and 0.025 and 0.975 (95 percent credible intervals).
Details

type "relative" is calculated as the percentage decrease from the maximum predicted value of the response (top) to the minimum predicted value of the response. Type "absolute" (the default) is calculated as the percentage decrease from the maximum value of the response (top) to 0 (or both for a 4 parameter model fit). Type "direct" provides a direct estimate of the x value for a given y. Note that for the current version, ECx for an "nechorme" (NEC Hormesis) model is estimated at a percent decline from the control. For hormesis_def, if "max", then ECx values are calculated as a decline from the maximum estimates (i.e. the peak at nec); if "control", then ECx values are calculated relative to the control, which is assumed to be the lowest observed concentration.

Value

A vector containing the estimated ECx value, including upper and lower 95% credible interval bounds.

See Also

bneec

Examples

library(brms)
library(bayesnec)
data(manec_example)
ecx(manec_example, ecx_val = 50)
ecx(manec_example)

Description

Extracts the predicted ECx value as desired from an object of class bayesmanecfit.

Usage

## S3 method for class 'bayesmanecfit'
ecx(
  object,
  ecx_val = 10,
  precision = 1000,
  posterior = FALSE,
  type = "absolute",
  hormesis_def = "control",
  x_range = NA,
ecx.bayesmanecfit

xform = NA,
    prob_vals = c(0.5, 0.025, 0.975)
)

Arguments

object       An object of class bayesmanecfit returned by bnec.
ecx_val      The desired percentage effect value. This must be a value between 1 and 99 (for
             type = "relative" and "absolute"), defaults to 10.
precision    The number of unique x values over which to find ECx - large values will make
             the ECx estimate more precise.
posterior    A logical value indicating if the full posterior sample of calculated ECx values
             should be returned instead of just the median and 95 credible intervals.
type         A character vector, taking values of "relative", "absolute" (the default) or "di-
             rect". See Details.
hormesis_def A character vector, taking values of "max" or "control". See Details.
x_range      A range of x values over which to consider extracting ECX.
xform        A function to apply to the returned estimated concentration values.
prob_vals    A vector indicating the probability values over which to return the estimated
             ECX value. Defaults to 0.5 (median) and 0.025 and 0.975 (95 percent credible
             intervals).

Details

type "relative" is calculated as the percentage decrease from the maximum predicted value of the
response (top) to the minimum predicted value of the response. Type "absolute" (the default) is
calculated as the percentage decrease from the maximum value of the response (top) to 0 (or bot
for a 4 parameter model fit). Type "direct" provides a direct estimate of the x value for a given y.
Note that for the current version, ECX for an "nechorme" (NEC Hormesis) model is estimated at
a percent decline from the control. For hormesis_def, if "max", then ECX values are calculated
as a decline from the maximum estimates (i.e. the peak at nec); if "control", then ECX values are
calculated relative to the control, which is assumed to be the lowest observed concentration.

Value

A vector containing the estimated ECX value, including upper and lower 95% credible interval
bounds.

See Also

bnec

Examples

library(brms)
library(bayesnec)
data(manec_example)
ecx(manec_example, ecx_val = 50)
ecx(manec_example)

description
Extracts the predicted ECx value as desired from an object of class \texttt{bayesnecfit}.

Usage

\begin{verbatim}
## S3 method for class 'bayesnecfit'
ecx(object, ...)
\end{verbatim}

Arguments

- \texttt{object}: An object of class \texttt{bayesnecfit} returned by \texttt{bnec}.
- \texttt{...}: Additional arguments to \texttt{ecx}

Details

type "relative" is calculated as the percentage decrease from the maximum predicted value of the response (top) to the minimum predicted value of the response. Type "absolute" (the default) is calculated as the percentage decrease from the maximum value of the response (top) to 0 (or bot for a 4 parameter model fit). Type "direct" provides a direct estimate of the x value for a given y. Note that for the current version, ECx for an "nechorme" (NEC Hormesis) model is estimated at a percent decline from the control. For \texttt{hormesis_def}, if "max", then ECx values are calculated as a decline from the maximum estimates (i.e. the peak at nec); if "control", then ECx values are calculated relative to the control, which is assumed to be the lowest observed concentration.

Value

A vector containing the estimated ECx value, including upper and lower 95\% credible interval bounds.

See Also

\texttt{bnec}
 Examples

```r
library(brms)
library(bayesnec)
data(manec_example)
ecx(manec_example, ecx_val = 50)
ecx(manec_example)
```

---

## ecx.default

### Description

Extracts the predicted ECx value as desired from an object of class `bayesnecfit` or `bayesmanecfit`.

### Usage

```r
## Default S3 method:
ecx(
  object,
  ecx_val = 10,
  precision = 1000,
  posterior = FALSE,
  type = "absolute",
  hormesis_def = "control",
  x_range = NA,
  xform = NA,
  prob_vals = c(0.5, 0.025, 0.975)
)
```

### Arguments

- **object**
  - An object of class `bayesnecfit` or `bayesmanecfit` returned by `bnec`.
- **ecx_val**
  - The desired percentage effect value. This must be a value between 1 and 99 (for type = "relative" and "absolute"), defaults to 10.
- **precision**
  - The number of unique x values over which to find ECx - large values will make the ECx estimate more precise.
- **posterior**
  - A logical value indicating if the full posterior sample of calculated ECx values should be returned instead of just the median and 95 credible intervals.
- **type**
  - A character vector, taking values of "relative", "absolute" (the default) or "direct". See Details.
- **hormesis_def**
  - A character vector, taking values of "max" or "control". See Details.
- **x_range**
  - A range of x values over which to consider extracting ECx.
xform  A function to apply to the returned estimated concentration values.
prob_vals  A vector indicating the probability values over which to return the estimated ECx value. Defaults to 0.5 (median) and 0.025 and 0.975 (95 percent credible intervals).

Details

type "relative" is calculated as the percentage decrease from the maximum predicted value of the response (top) to the minimum predicted value of the response. Type "absolute" (the default) is calculated as the percentage decrease from the maximum value of the response (top) to 0 (or bot for a 4 parameter model fit). Type "direct" provides a direct estimate of the x value for a given y. Note that for the current version, ECx for an "nechorme" (NEC Hormesis) model is estimated at a percent decline from the control. For hormesis_def, if "max", then ECx values are calculated as a decline from the maximum estimates (i.e. the peak at nec); if "control", then ECx values are calculated relative to the control, which is assumed to be the lowest observed concentration.

Value

A vector containing the estimated ECx value, including upper and lower 95% credible interval bounds.

See Also

bnec

Examples

library(brms)
library(bayesnec)
data(manec_example)
ecx(manec_example, ecx_val = 50)
ecx(manec_example)

Description

Extracts a range of statistics from a list of bayesnecfit model fits.
expand_nec

Usage

expand_manec(
  object,
  x_range = NA,
  precision = 1000,
  sig_val = 0.01,
  loo_controls = list(method = "pseudobma")
)

Arguments

object a bayesmanecfit mod_fits output list, as returned by bnc when more than one model is supplied.
x_range A range of x values over which to consider extracting ECx.
precision The length of the x vector used for posterior predictions, and over which to extract ECx values. Large values will be slower but more precise.
sig_val Probability value to use as the lower quantile to test significance of the predicted posterior values against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECx curves.
loo_controls A named list containing the desired arguments to be passed on to loo_model_weights. It sets the default wi_method to "pseudobma". See help documentation ?loo_model_weights from package loo.

Value

A list of model statistical output derived from the input model list

Description

Assigns class to a prebayesncfit object

Usage

expand_nec(object, x_range = NA, precision = 1000, sig_val = 0.01)

Arguments

object An object of class prebayesncfit.
x_range A range of x values over which to consider extracting ECx.
precision The length of the x vector used for posterior predictions, and over which to extract ECx values. Large values will be slower but more precise.
**Value**

An object of class bayesnecfit.

---

**extract_simdat**

**Description**

Extracts a range of statistics from a list of bayesnecfit model fits.

**Usage**

`extract_simdat(modfit)`

**Arguments**

- `modfit`: a bayesnecfit, as returned by `fit_bayesnec`

**Value**

A list of model simulation statistics including iter, thin, warmup and chains

---

**fit_bayesnec**

**Description**

Fits a concentration(dose)-response model using brms.

**Usage**

```r
fit_bayesnec(
  data,
  x_var,
  y_var,
  trials_var = NA,
  family = NULL,
  priors,
  model = NA,
  inits,
```
```r
skip_check = FALSE,
pointwise,
random = NA,
random_vars = NA,
...
)

Arguments

data A `data.frame` containing the data to use for the model.
x_var A `character` indicating the column heading containing the concentration (x) variable.
y_var A `character` indicating the column heading containing the response (y) variable.
trials_var The column heading indicating the column for the number of "trials" for binomial or beta_binomial2 response data. If not supplied, the model may run but will not be the model you intended!
family Either a `character` string, a function, or an object of class `family` defining the statistical distribution (family) to use for the y (response) data. See details.
priors An object of class `brmsprior` which specifies user-desired prior distributions of model parameters. If missing, `bnc` will figure out a baseline prior for each parameter. It can also be specified as a named `list` where each name needs to correspond to the same string as "model". See details.
model A `character` vector indicating the model(s) to fit. See Details for more information.
inits Optional. Initialisation values. Must be a `list` of "n" names lists, where "n" corresponds to the number of chains, and names correspond to the parameter names of a given model.
skip_check Should data check via `check_data` be avoided? Only relevant to function `amend`. Defaults to FALSE.
pointwise A flag indicating whether to compute the full log-likelihood matrix at once or separately for each observation. The latter approach is usually considerably slower but requires much less working memory. Accordingly, if one runs into memory issues, `pointwise = TRUE` is the way to go, but will not work for the custom family beta_binomial2
random = A named `list` containing the random model formula to apply to model parameters.
random_vars = A `character` vector containing the names of the columns containing the variables used in the random model formula.

... Further arguments to `brm` via `fit_bayesnec`.

Value

The fitted `brms` model, including an estimate of the NEC value and predicted posterior values. A posterior sample of the NEC is also available under `nec_posterior`
See Also

bpec

Description

bayesnec standard ggplot method.

Usage

ggbnec(x, nec = TRUE, ecx = FALSE, ...)

Arguments

x  An object of class bayesnecfit or bayesmanecfit, as returned by function
    bpec.

nec  Should NEC values be added to the plot? Defaults to TRUE.

ecx  Should ECx values be added to the plot? Defaults to FALSE.

...  Arguments passed on to ggbnec.bayesmanecfit

all  Should all individual models be plotted separately (defaults to TRUE) or
    should model averaged predictions be plotted instead?

plot  Should output ggplot output be plotted? Only relevant if all = TRUE and
    multi_facet = FALSE.

ask  Indicates if the user is prompted before a new page is plotted. Only relevant
    if plot = TRUE and multi_facet = FALSE.

newpage  Indicates if the first set of plots should be plotted to a new page. Only
    relevant if plot = TRUE and multi_facet = FALSE.

multi_facet  Should all plots be plotted in one single panel via facets? De-
    faults to FALSE.

Value

A ggplot object.

Examples

library(brms)
library(bayesnec)
options(mc.cores = 2)
data(nec_data)

test <- bpec(data = nec_data, x_var = "x", y_var = "y",
    model = c("nec3param", "nec4param"), iter = 2e2,
    ...)
family = Beta(link = "identity")

# plots multiple models, one at a time, with interactive prompt
# plot model averaged predictions
# plot all panels together

ggbnec(test, ecx = TRUE, ecx_val = 50, multi_facet = TRUE)
ask Indicates if the user is prompted before a new page is plotted. Only relevant if plot = TRUE and multi_facet = FALSE.

newpage Indicates if the first set of plots should be plotted to a new page. Only relevant if plot = TRUE and multi_facet = FALSE.

multi_facet Should all plots be plotted in one single panel via facets? Defaults to FALSE.

Value

A ggplot object.

Examples

```r
library(bayesnec)
data(manec_example)

ggbnec_data(manec_example)
ggbnec_data(manec_example, add_ecx = TRUE, ecx_val = 50)
```

Description

bayesnec standard ggplot method.

Usage

```r
## S3 method for class 'bayesnecfit'
ggbnec(x, nec = TRUE, ecx = FALSE, ...)
```

Arguments

- `x` An object of class bayesnecfit as returned by function bnec.
- `nec` Should NEC values be added to the plot? Defaults to TRUE.
- `ecx` Should ECx values be added to the plot? Defaults to FALSE.
- `...` Additional arguments to be passed to ggbnec_data.

Value

A ggplot object.
Examples

```r
library(bayesnec)
data(manec_example)

ggbnec_data(manec_example)

ggbnec_data(manec_example, add_ecx = TRUE, ecx_val = 50)
```

Description

`bayesnec` standard ggplot method.

Usage

```r
## Default S3 method:
ggbnec(x, nec = TRUE, ecx = FALSE, ...)
```

Arguments

- **x**: A data.frame created by function `ggbnec_data`.
- **nec**: Should NEC values be added to the plot? Defaults to TRUE.
- **ecx**: Should ECx values be added to the plot? Defaults to FALSE.
- **...**: Unused.

Value

A `ggplot` object.

Examples

```r
library(bayesnec)
data(manec_example)

ggbnec_data(manec_example)

ggbnec_data(manec_example, add_ecx = TRUE, ecx_val = 50)
```
ggbnec_data

Description

Creates the data.frame that feeds into the plotting function `ggbnec`.

Usage

```r
 ggbnec_data(x, add_nec = TRUE, add_ecx = FALSE, ...)
```

Arguments

- `x`: An object of class `bayesnecfit` or `bayesmanecfit`, as returned by function `bnec`.
- `add_nec`: Should NEC values be added to the plot? Defaults to TRUE.
- `add_ecx`: Should ECx values be added to the plot? Defaults to FALSE.
- `...`: Additional arguments to be passed to `ecx`. By default, function `ecx` returns EC10.

Value

A `data.frame`.

Examples

```r
library(bayesnec)
data(manec_example)

ggbnec_data(manec_example)
ggbnec_data(manec_example, add_ecx = TRUE, ecx_val = 50)
```
**Usage**

```r
## S3 method for class 'bayesmanecfit'
ggbnec_data(x, add_nec = TRUE, add_ecx = FALSE, ...)
```

**Arguments**

- `x`: An object of class `bayesmanecfit`, as returned by function `bnc`.  
- `add_nec`: Should NEC values be added to the plot? Defaults to TRUE.  
- `add_ecx`: Should ECx values be added to the plot? Defaults to FALSE.  
- `...`: Additional arguments to be passed to `ecx`. By default, function `ecx` returns EC10.

**Value**

A `data.frame`.

**Examples**

```r
library(bayesnec)
data(manec_example)

ggbnec_data(manec_example)
ggbnec_data(manec_example, add_ecx = TRUE, ecx_val = 50)
```

---

**Description**

Creates the `data.frame` that feeds into the plotting function `ggbnc`.

**Usage**

```r
## S3 method for class 'bayesnecfit'
ggbnec_data(x, ...)```

**Arguments**

- `x`: An object of class `bayesnecfit`, as returned by function `bnc`.  
- `...`: Additional arguments to be passed to `ecx`. By default, function `ecx` returns EC10.
Value

A `data.frame`.

Examples

```{r}
library(bayesnec)
data(manec_example)

ggbnec_data(manec_example)
ggbnec_data(manec_example, add_ecx = TRUE, ecx_val = 50)
```

---

### ggbnec_data.default

**Description**

Creates the `data.frame` that feeds into the plotting function `ggbnec`.

**Usage**

```r
## Default S3 method:
ggbnec_data(x, add_nec = TRUE, add_ecx = FALSE, ...)
```

**Arguments**

- `x` An object of class `bayesnecfit`, as returned by function `bnec`.
- `add_nec` Should NEC values be added to the plot? Defaults to `TRUE`.
- `add_ecx` Should ECx values be added to the plot? Defaults to `FALSE`.
- `...` Additional arguments to be passed to `ecx`. By default, function `ecx` returns `EC10`.

**Value**

A `data.frame`.

**Examples**

```{r}
library(bayesnec)
data(manec_example)

ggbnec_data(manec_example)
ggbnec_data(manec_example, add_ecx = TRUE, ecx_val = 50)
```
is_bayesmanecfit

Description
Checks if argument is a bayesmanecfit object

Usage
is_bayesmanecfit(x)

Arguments
x An R object

Value
A logical

is_bayesnecfit

Description
Checks if argument is a bayesnecfit object

Usage
is_bayesnecfit(x)

Arguments
x An R object

is_manecsummary

Description
Checks if argument is a manecsummary object

Usage
is_manecsummary(x)

Arguments
x An R object

Value
A logical
is_necsummary  Checks if argument is a necsummary object

Description
Checks if argument is a necsummary object

Usage
is_necsummary(x)

Arguments
x  An R object

is_prebayesnecfit  Checks if argument is a prebayesnecfit object

Description
Checks if argument is a prebayesnecfit object

Usage
is_prebayesnecfit(x)

Arguments
x  An R object

Value
A logical vector.
**Description**

Beta-binomial wrapper LL.

**Usage**

\[
\text{log\_lik\_beta\_binomial2}(i, \text{prep})
\]

**Arguments**

- \(i\): observation \(i\).
- \(\text{prep}\): data with posterior.

**Value**

Log likelihood of the beta binomial distribution

---

**make_good_inits**

**Description**

Creates list of initialisation values that generate data within the natural range of data.

**Usage**

\[
\text{make\_good\_inits}(\text{model}, x, y, \text{n\_trials} = 1e+05, \ldots)
\]

**Arguments**

- \(\text{model}\): A character vector indicating the model(s) to fit. See Details for more information.
- \(x\): A numeric vector containing the \(x\) predictor.
- \(y\): A numeric vector containing the \(y\) response.
- \(\text{n\_trials}\): A numeric vector indicating how many attempts the function should run before giving up.
- \(...\): Additional arguments to \text{make\_inits}.

**Value**

A list containing the initialisation values.
See Also

make_inits

make_inits

Description

Creates list of initialisation values

Usage

make_inits(model, fct_args, priors, chains)

Arguments

model A character vector indicating the model(s) to fit. See Details for more information.
fct_args A character string containing the expected argument names to be used.
priors an object of class "brmsprior" from package brms.
chains Number of chains to be passed to brms model.

Value

A list containing the initialisation values.

See Also

bnec

Class manecsummary of models fitted with the brms package

Description

Multiple models fitted with the bayesnec package are summarised as a manecsummary object, which contains the name of the non-linear models fitted, the family distribution used to fit all the models, the total post-warm-up sample size, a table containing the model weights, the method to calculate the weights, whether this model is an ECx-type model (see details below), and the ECx summary values should the user decide to calculate them.

Details

See methods(class = "manecsummary") for an overview of available methods.
Slots

- models: A character string indicating the name of the fitted non-linear models.
- family: The family distribution used to fit all the models.
- sample_size: The total post-warm-up sample size.
- mod_weights: A table containing the model weights.
- mod_weights_method: The method to calculate the weights.
- ecx_mods: A logical indicating which models are ECx-type models.
- nec_vals: The model-averaged NEC values. Note that if model stack contains ECx-type models, these will be via NSEC proxies.
- ecs: A list containing the ECx values should the user decide to calculate them (see the non-exported bayesnec:::summary.bayesnecfit help file for details). Different from the single-model case of class bayesnecfit, these ECx estimates will be based on the model weights.
- rhat_issues: A list detailing whether each fitted model exhibited convergence issues based on the Rhat evaluation.

See Also

bayesnec, bnec, bayesnecfit, bayesmanecfit, necsummary

---

**manec_example**

*Example bayesmanecfit object*

**Description**

Example bayesmanecfit object

**Format**

An object of class bayesmanecfit. This was created to reduce run time in examples and tests, and to give the user an example to toy with. This was fitted to bayesnec built-in mock dataset (see ?nec_data), using models "nec4param" and "ecx4param". The number of chains were set to 2 and number of iterations were 50 only to make sure that package size was below 5 Mb. See help files for function bnec and class bayesmanecfit for details.

**Source**

Code used to generate these models can be downloaded from https://github.com/open-AIMS/bayesnec/blob/master/data-raw/manec_example.R
Description

Lists the fitted or available models

Usage

`models(object)`

Arguments

- `object` An object of class `bayesnecfit` or `bayesmanecfit` as returned by `bnec`, a character vector indicating the type of model set for which to list the available models, or a numeric vector indicating the natural range of values which the models should be able to handle (see details). If missing all available models and their groups are listed.

Details

The available models are "ecx4param", "ecxexp", "ecxlin", "ecxsigmoid", "ecxwb1", "ecxwb2", "nec3param", "nec4param", "nec horme", "nec horme4", "neclin", "neclin horme" and "nec sigmoid".

To see the model formula and parameters for a specific model use the function `show_params`.

To see all the models in an available set (e.g. "all", "nec" or ecx") use the function `models` specifying the group name.

To see the model names, model formula and parameters fitted in an existing bayesnec or bayesmanec model object use the function `models` specifying the fitted object.

To see what models are available for a given type of data use the function `models` passing a numeric vector indicating the range of possible data types. Models that have an exponential decay (most models with parameter "beta") with no "bot" parameter are zero_bounded and are not suitable for the gaussian family, or any family modelled using a logit or log link function. Models with a linear decay (containing the string "lin" in their name) are not suitable for modelling families that are zero bounded (gamma, poisson, negative binomial) using an identity link. Models with a linear decay or hormesis linear increase (all models with parameter "slope") are not suitable # for modelling families that are 0, 1 bounded (binomial, beta, betabinomial2). These restrictions do not need to be controlled by the user and a call to `bnec` with `models = "all"` will simply exclude inappropriate models.

Value

A list of the available or fitted models
Examples

```r
library(bayesnec)
# default to all models and model groups
models()
# single model
show_params("nec3param")
# group of models
models("all")
# models that are suitable for 0,1 bounded data
models(c(0,1))
```

Description

Extracts the predicted nec value as desired from an object of class `bayesnecfit` or `bayesnecfit`.

Usage

```r
nec(object, posterior = FALSE, xform = NA, prob_vals = c(0.5, 0.025, 0.975))
```

Arguments

- `object` - An object of class `bayesnecfit` or `bayesnecfit` returned by `bnc`.
- `posterior` - A `logical` value indicating if the full posterior sample of calculated nec values should be returned instead of just the median and 95 credible intervals.
- `xform` - A function to apply to the returned estimated concentration values.
- `prob_vals` - A vector indicating the probability values over which to return the estimated nec value. Defaults to 0.5 (median) and 0.025 and 0.975 (95 percent credible intervals).

Value

A vector containing the estimated nec value, including upper and lower 95% credible interval bounds (or other interval as specified by `prob_vals`).

See Also

- `bnc`
nec.bayesmanecfit

Description

Extracts the predicted nec value as desired from an object of class `bayesmanecfit`.

Usage

```r
## S3 method for class 'bayesmanecfit'
nec(object, posterior = FALSE, xform = NA, prob_vals = c(0.5, 0.025, 0.975))
```

Arguments

- `object`: An object of class `bayesmanecfit` returned by `bnec`.
- `posterior`: A logical value indicating if the full posterior sample of calculated nec values should be returned instead of just the median and 95 credible intervals.
- `xform`: A function to apply to the returned estimated concentration values.
- `prob_vals`: A vector indicating the probability values over which to return the estimated nec value. Defaults to 0.5 (median) and 0.025 and 0.975 (95 percent credible intervals).

Value

A vector containing the estimated nec value, including upper and lower 95% credible interval bounds (or other interval as specified by `prob_vals`).

See Also

- `bnec`

Examples

```r
library(bayesnec)
data(manec_example)
nec(manec_example)
```
Description

Extracts the predicted nec value as desired from an object of class `bayesnecfit`.

Usage

```r
## S3 method for class 'bayesnecfit'
nec(object, ...)
```

Arguments

- `object`: An object of class `bayesnecfit` returned by `bnec`.
- `...`: Additional arguments to `nec`

Value

A vector containing the estimated nec value, including upper and lower 95% credible interval bounds (or other interval as specified by `prob_vals`).

See Also

`bnec`

Examples

```r
library(bayesnec)
data(manec_example)
nec(manec_example)
```

Description

Extracts the predicted nec value as desired from an object of class `bayesnecfit` or `bayesnecfit`.

Usage

```r
## Default S3 method:
nec(object, posterior = FALSE, xform = NA, prob_vals = c(0.5, 0.025, 0.975))
```
necsummary-class

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>An object of class <code>bayesnecfit</code> or <code>bayesmanecfit</code> returned by <code>bnec</code>.</td>
</tr>
<tr>
<td>posterior</td>
<td>A logical value indicating if the full posterior sample of calculated nec values should be returned instead of just the median and 95 credible intervals.</td>
</tr>
<tr>
<td>xform</td>
<td>A function to apply to the returned estimated concentration values.</td>
</tr>
<tr>
<td>prob_vals</td>
<td>A vector indicating the probability values over which to return the estimated nec value. Defaults to 0.5 (median) and 0.025 and 0.975 (95 percent credible intervals).</td>
</tr>
</tbody>
</table>

Value

A vector containing the estimated nec value, including upper and lower 95% credible interval bounds (or other interval as specified by `prob_vals`).

See Also

`bnec`

Examples

```r
library(bayesnec)
data(manec_example)
nec(manec_example)
```

Description

Single models fitted with the `bayesnec` package are summarised as a `necsummary` object, which contains the original `brmsfit` object summary, the name of the non-linear model fitted, whether this model is an ECx-type model (see details below), and the ECx summary values should the user decide to calculate them.

Details

See `methods(class = "necsummary")` for an overview of available methods.

Slots

- `brmssummary` The standard summary for the fitted Bayesian model of class `brmsfit`.
- `model` A character string indicating the name of the fitted non-linear model.
- `is_ecx` A logical indicating whether model is an ECx-type model.
- `ecs` A list containing the ECx values should the user decide to calculate them (see the non-exported `bayesnec::summary.bayesnecfit` help file for details).
**nec_data**

*Example data of non-linear decay*

**Description**

A simulated dataset containing a series of response measurements as a function of a concentration axis. Data simulated by Diego Barneche.

**Format**

A data frame with 100 rows and 2 variables:

- **x**: Concentration (predictor) axis.
- **y**: Response.

**nsec**

**Description**

Extracts the predicted nsec value as desired from an object of class `bayesnecfit` or `bayesmanecfit`.

**Usage**

```r
nsec(
  object,
  sig_val = 0.01,
  precision = 1000,
  posterior = FALSE,
  x_range = NA,
  hormesis_def = "control",
  xform = NA,
  prob_vals = c(0.5, 0.025, 0.975)
)
```
Arguments

- **object**: An object of class `bayesnecfit` or `bayesnecfit` returned by `bnec`.
- **sig_val**: Probability value to use as the lower quantile to test significance of the predicted posterior values. Against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECx curves.
- **precision**: The number of unique x values over which to find nsec - large values will make the nsec estimate more precise.
- **posterior**: A logical value indicating if the full posterior sample of calculated nsec values should be returned instead of just the median and 95 credible intervals.
- **x_range**: A range of x values over which to consider extracting nsec.
- **hormesis_def**: A character vector, taking values of "max" or "control". See Details.
- **xform**: A function to apply to the returned estimated concentration values.
- **prob_vals**: A vector indicating the probability values over which to return the estimated nsec value. Defaults to 0.5 (median) and 0.025 and 0.975 (95 percent credible intervals).

Details

For `hormesis_def`, if "max", then nsec values are calculated as a decline from the maximum estimates (i.e. the peak at nec); if "control", then ECx values are calculated relative to the control, which is assumed to be the lowest observed concentration.

Value

A vector containing the estimated nsec value, including upper and lower 95% credible interval bounds.

See Also

- `bnec`

Examples

```r
library(bayesnec)
data(manec_example)
nsec(manec_example)
```
Description

Extracts the predicted nsec value as desired from an object of class `bayesmanecfit`.

Usage

```r
## S3 method for class 'bayesmanecfit'

nsec(
  object,
  sig_val = 0.01,
  precision = 1000,
  posterior = FALSE,
  x_range = NA,
  hormesis_def = "control",
  xform = NA,
  prob_vals = c(0.5, 0.025, 0.975)
)
```

Arguments

- `object`: An object of class `bayesmanecfit` returned by `bnec`.
- `sig_val`: Probability value to use as the lower quantile to test significance of the predicted posterior values, against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECx curves.
- `precision`: The number of unique x values over which to find nsec - large values will make the nsec estimate more precise.
- `posterior`: A logical value indicating if the full posterior sample of calculated nsec values should be returned instead of just the median and 95 percent credible intervals.
- `x_range`: A range of x values over which to consider extracting nsec.
- `hormesis_def`: A character vector, taking values of "max" or "control". See Details.
- `xform`: A function to apply to the returned estimated concentration values.
- `prob_vals`: A vector indicating the probability values over which to return the estimated nsec value. Defaults to 0.5 (median) and 0.025 and 0.975 (95 percent credible intervals).

Details

For `hormesis_def`, if "max", then nsec values are calculated as a decline from the maximum estimates (i.e. the peak at nec); if "control", then ECx values are calculated relative to the control, which is assumed to be the lowest observed concentration.
Value

A vector containing the estimated nsec value, including upper and lower 95% credible interval bounds.

See Also

bnec

Examples

library(bayesnec)

data(manec_example)
nsec(manec_example)

Description

Extracts the predicted nsec value as desired from an object of class bayesnecfit.

Usage

## S3 method for class 'bayesnecfit'
nsec(object, ...)

Arguments

object An object of class bayesnecfit returned by bnec.

... Additional arguments to nsec

Details

For hormesis_def, if "max", then nsec values are calculated as a decline from the maximum estimates (i.e. the peak at nec); if "control", then ECx values are calculated relative to the control, which is assumed to be the lowest observed concentration.

Value

A vector containing the estimated nsec value, including upper and lower 95% credible interval bounds.

See Also

bnec
Examples

```r
library(bayesnec)

data(manec_example)
nsec(manec_example)
```

### Description

Extracts the predicted nsec value as desired from an object of class `bayesnecfit` or `bayesmanecfit`.

### Usage

```r
## Default S3 method:
nsec(
  object,
  sig_val = 0.01,
  precision = 1000,
  posterior = FALSE,
  x_range = NA,
  hormesis_def = "control",
  xform = NA,
  prob_vals = c(0.5, 0.025, 0.975)
)
```

### Arguments

- **object**: An object of class `bayesnecfit` or `bayesmanecfit` returned by `bnec`.
- **sig_val**: Probability value to use as the lower quantile to test significance of the predicted posterior values, against the lowest observed concentration (assumed to be the control), to estimate NEC as an interpolated NOEC value from smooth ECx curves.
- **precision**: The number of unique x values over which to find nsec - large values will make the nsec estimate more precise.
- **posterior**: A **logical** value indicating if the full posterior sample of calculated nsec values should be returned instead of just the median and 95 percent credible intervals.
- **x_range**: A range of x values over which to consider extracting nsec.
- **hormesis_def**: A **character** vector, taking values of "max" or "control". See Details.
- **xform**: A function to apply to the returned estimated concentration values.
- **prob_vals**: A vector indicating the probability values over which to return the estimated nsec value. Defaults to 0.5 (median) and 0.025 and 0.975 (95 percent credible intervals).
Details

For `hormesis_def`, if "max", then nsec values are calculated as a decline from the maximum estimates (i.e. the peak at nec); if "control", then ECx values are calculated relative to the control, which is assumed to be the lowest observed concentration.

Value

A vector containing the estimated nsec value, including upper and lower 95% credible interval bounds.

See Also

`bnec`

Examples

```r
library(bayesnec)

data(manec_example)
nsec(manec_example)
```

```
plot.bayesmanecfit  plot.bayesmanecfit
```

Description

Generates a plot of a fitted `bayesmanecfit` object, as returned by `bnec`.

Usage

```r
## S3 method for class 'bayesmanecfit'
plot(
  x,
  ..., 
  CI = TRUE,
  add_nec = TRUE,
  position_legend = "topright",
  add_ec10 = FALSE,
  xform = NA,
  lxform = NA,
  jitter_x = FALSE,
  jitter_y = FALSE,
  ylab = "response",
  xlab = "concentration",
  xticks = NA,
)```
plot.bayesnecfit

```r
all_models = FALSE
```

### Arguments

- **x**
  - An object of class `bayesnecfit` as returned by `bnec`.
- **...**
  - Additional arguments to `plot`.
- **CI**
  - A `logical` value indicating if credibility intervals on the model fit should be plotted, calculated as the upper and lower bounds of the individual predicted values from all posterior samples.
- **add_nec**
  - A `logical` value indicating if the estimated NEC value and 95% credible intervals should be added to the plot.
- **position_legend**
  - A `numeric` vector indicating the location of the NEC or EC10 legend, as per a call to legend.
- **add_ec10**
  - A `logical` value indicating if an estimated EC10 value and 95% credible intervals should be added to the plot.
- **xform**
  - A function to be applied as a transformation of the x data.
- **lxform**
  - A function to be applied as a transformation only to axis labels and the annotated NEC / EC10 values.
- **jitter_x**
  - A `logical` value indicating if the x data points on the plot should be jittered.
- **jitter_y**
  - A `logical` value indicating if the y data points on the plot should be jittered.
- **ylab**
  - A `character` vector to use for the y-axis label.
- **xlab**
  - A `character` vector to use for the x-axis label.
- **xticks**
  - A numeric vector indicating where to place the tick marks of the x-axis.
- **all_models**
  - A `logical` value indicating if all models in the model set should be plotted simultaneously, or if a model average plot should be returned.

### Value

A plot of the fitted model

---

### Description

Generates a plot of a fitted "bayesnecfit" model, as returned by `bnec`. 
Usage

```r
## S3 method for class 'bayesnecfit'
plot(
  x,
  ...,
  CI = TRUE,
  add_nec = TRUE,
  position_legend = "topright",
  add_ec10 = FALSE,
  xform = NA,
  lxform = NA,
  jitter_x = FALSE,
  jitter_y = FALSE,
  ylab = "response",
  xlab = "concentration",
  xticks = NA
)
```

Arguments

- `x` : An object of class `bayesnecfit` as returned by `bnec`.
- `...` : Additional arguments to `plot`.
- `CI` : A `logical` value indicating if credibility intervals on the model fit should be plotted, calculated as the upper and lower bounds of the individual predicted values from all posterior samples.
- `add_nec` : A `logical` value indicating if the estimated NEC value and 95% credible intervals should be added to the plot.
- `position_legend` : A `numeric` vector indicating the location of the NEC or EC10 legend, as per a call to `legend`.
- `add_ec10` : A `logical` value indicating if an estimated EC10 value and 95% credible intervals should be added to the plot.
- `xform` : A function to be applied as a transformation of the x data.
- `lxform` : A function to be applied as a transformation only to axis labels and the annotated NEC / EC10 values.
- `jitter_x` : A `logical` value indicating if the x data points on the plot should be jittered.
- `jitter_y` : A `logical` value indicating if the y data points on the plot should be jittered.
- `ylab` : A `character` vector to use for the y-axis label.
- `xlab` : A `character` vector to use for the x-axis label.
- `xticks` : A numeric vector indicating where to place the tick marks of the x-axis.

Value

A plot of the fitted model.
**posterior_epred_beta_binomial2**

**Description**
Beta-binomial wrapper posterior_epred method

**Usage**
```r
posterior_epred_beta_binomial2(prep)
```

**Arguments**
- `prep`: data with posterior.

**Value**
A numeric value or vector containing predicted random values of the beta binomial distribution

---

**posterior_predict_beta_binomial2**

**Description**
Beta-binomial wrapper posterior_predict method

**Usage**
```r
posterior_predict_beta_binomial2(i, prep, ...)
```

**Arguments**
- `i`: observation i.
- `prep`: data with posterior.
- `...`: unused.

**Value**
A numeric value or vector containing predicted probability values of the beta binomial distribution
**prebayesnecfit-class**  
*Class prebayesnecfit of models fitted with the brms package*

**Description**
This is an intermediate class that was created to make both `bayesnecfit` and `bayesmanecfit` objects lighter to handle. It contains the original `brmsfit` fitted object, name of non-linear model that was fitted, and the list of initialisation values applied.

**Details**
See `methods(class = "prebayesnecfit")` for an overview of available methods.

**Slots**
- **fit**  
The fitted Bayesian model of class `brmsfit`.
- **model**  
A character string indicating the name of the fitted model.
- **inits**  
A list containing the initialisation values for to fit the model.

**See Also**
- `bayesnec`, `bnec`, `bayesnecfit`, `bayesmanecfit`

---

**predict.bayesmanecfit**

**Description**
`predict.bayesmanecfit`

**Usage**
```r
## S3 method for class 'bayesmanecfit'
predict(object, ..., precision = 100, x_range = NA)
```

**Arguments**
- **object**  
An object of class `bayesmanecfit` as returned by `bnec`.
- **...**  
Unused.
- **precision**  
the number of x values over which to predict values.
- **x_range**  
The range of x values over which to make predictions.

**Value**
A list containing x and fitted y, with up and lw values

---
predict.bayesnecfit

Description
predict.bayesnecfit

Usage
## S3 method for class 'bayesnecfit'
predict(object, ..., precision = 100, x_range = NA)

Arguments

object
An object of class bayesnecfit as returned by bnec.

... Unused.

precision
the number of x values over which to predict values.

x_range
The range of x values over which to make predictions.

Value
A list containing x and fitted y, with up and lw values

prep_raw_data

Description
prep_raw_data

Usage
prep_raw_data(brms_fit)

Arguments

brms_fit
A brmsfit object.

Value
A data.frame.
## print.bayesnecfit

### Description

print.bayesnecfit

### Usage

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

### Arguments

- `x`  
  An object of class `bayesnecfit` as returned by `bnec`.

- `...`  
  Further arguments to function `summary`.

### Value

A list containing a summary of the model fit as returned a `brmsfit` for each model.

---

## print.bayesmanecfit

### Description

print.bayesmanecfit

### Usage

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

### Arguments

- `x`  
  An object of class `bayesmanecfit` as returned by `bnec`.

- `...`  
  Further arguments to function `summary`.

### Value

A list containing a summary of the model fit as returned a `brmsfit` for each model.
### Description
print.manecsummary

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

### Arguments
- `x` An object of class `manecsummary` as returned by `summary.bayesmanecfit`.
- `...` Unused.

### Value
A list containing a summary of model features and statistics.

### Description
print.necsummary

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

### Arguments
- `x` An object of class `necsummary` as returned by `summary.bayesnecfit`.
- `...` Unused.

### Value
A list containing a summary of model features and statistics.
pull_out

Description

Pulls a single model from an existing bayesmanecfit object, and converts into a bayesnecfit object.

Usage

pull_out(manec, model, ...)

Arguments

manec  An object of class bayesmanecfit output list as returned by bnec.
model  A character string indicating which model or class of models to pull out.
...    Additional arguments to expand_nec.

Value

An object of class bayesnecfit.

Examples

library(bayesnec)
data(manec_example)
ecx4param <- pull_out(manec_example, model = "ecx4param")

pull_prior

Description

Extracts the priors from an object of class bayesnecfit or bayesmanecfit.

Usage

pull_prior(object)

Arguments

object  An object of class bayesnecfit or bayesmanecfit returned by bnec.

Value

A list containing the priors
Examples

```r
library(bayesnec)
data(manec_example)
pull_prior(manec_example)
```

Description

**rhat.bayesmanecfit**

Usage

```r
## S3 method for class 'bayesmanecfit'
rhat(object, rhat_cutoff = 1.05, ...)
```

Arguments

- `object` An object of class `bayesmanecfit` as returned by `bnec`.
- `rhat_cutoff` A numeric vector indicating the rhat criteria used to test for model convergence.
- `...` Unused.

Value

A list containing a vector or rhat values as returned for a brm fit for each parameter, for each of the fitted models.

**rhat.bayesnecfit**

Description

**rhat.bayesnecfit**

Usage

```r
## S3 method for class 'bayesnecfit'
rhat(object, ...)
```

Arguments

- `object` An object of class `bayesnecfit` as returned by `bnec`.
- `...` Unused.
Value

A named vector containing Rhat values as returned for a brmsfit for each of the estimated parameters.

---

**sample_priors**

Description

Creates list or generates a plot of prior samples

Usage

```r
sample_priors(priors, n_samples = 10000, plot = "ggplot")
```

Arguments

- `priors`: an object of class "brmsprior" from package `brms`.
- `n_samples`: the number of prior samples to return.
- `plot`: NA returns a list of numeric vectors of sampled priors, "ggplot" (default) returns a ggplot and "base" returns a histogram in base R.

Value

A list containing the initialisation values.

See Also

- `bnec`

---

**set_distribution**

Sets distribution based on vector

Description

Sets distribution based on vector

Usage

```r
set_distribution(x, support_integer = FALSE, trials = NULL)
```

Arguments

- `x`: A numeric vector.
- `support_integer`: Logical. Should integer vectors be supported? Defaults to FALSE.
- `trials`: A numeric vector containing the number of trials. Defaults to NULL.
show_params

Details

Checks a vector and recommends a family distribution.

Value

A character vector.

Description

Displays non-linear equation and parameter names

Usage

show_params(model = "all")

Arguments

model A character vector indicating the model(s) to fit. See Details for more information.

Value

An object of class brmsformula.

Examples

library(bayespec)
# default to all models (i.e. model = "all")
show_params()
# single model
show_params(model = "nec3param")
# group of models
show_params(model = c("nec3param", "ecx"))
## S3 method for class 'bayesmanecfit'
summary(object, ..., ecx = FALSE, ecx_vals = c(10, 50, 90))

### Arguments
- **object**: An object of class `bayesmanecfit` as returned by `bnec`.
- **...**: Unused.
- **ecx**: Should summary EC values be calculated? Defaults to FALSE.
- **ecx_vals**: EC targets (between 1 and 99). Only relevant if ecx = TRUE. If no value is specified by the user, returns calculations for EC10, EC50, and EC90.

### Value
A list containing a summary of the model fit as returned a brmsfit for each model.

---

## S3 method for class 'bayesnecfit'
summary(object, ..., ecx = FALSE, ecx_vals = c(10, 50, 90))

### Arguments
- **object**: An object of class `bayesnecfit` as returned by `bnec`.
- **...**: Unused.
- **ecx**: Should summary EC values be calculated? Defaults to FALSE.
- **ecx_vals**: EC targets (between 1 and 99). Only relevant if ecx = TRUE. If no value is specified by the user, returns calculations for EC10, EC50, and EC90.

### Value
A summary of the fitted model as returned for a brmsfit
validate_family

**Description**

Checks if family is allowed

**Usage**

validate_family(family)

**Arguments**

family  Either a character string, a function, or an object of class family defining the statistical distribution (family) to use for the y (response) data. See details.

**Value**

An object of class family.

validate_priors

**Description**

Checks if priors are appropriate to specified model

**Usage**

validate_priors(priors, model)

**Arguments**

priors  An object of class brmsprior which specifies user-desired prior distributions of model parameters. If missing, bnec will figure out a baseline prior for each parameter. It can also be specified as a named list where each name needs to correspond to the same string as "model". See details.

model  A character vector indicating the model(s) to fit. See Details for more information.

**Value**

A data.frame of class brmsprior.
Index

amend, 4, 37
amend.bayesmanecfit, 5
amend.default, 7
average_endpoints, 8

bayesmanecfit, 4–8, 11, 16–19, 21, 22, 24, 29–31, 33, 38, 39, 42, 43, 49, 50, 52, 54, 55, 57, 59, 60, 64, 66, 68, 69, 72
bayesmanecfit (bayesmanecfit-class), 10
bayesmanecfit-class, 10
bayesnec, 10, 11, 38–41, 48, 49, 54, 55, 64
bayesnec (bayesnec-package), 4
bayesnec-package, 4

bayesnecfit (bayesnecfit-class), 10
bayesnecfit-class, 10
beta_binomial2, 11
beta_binomial2_lpmf, 12
beta_binomial2_rng, 12
bind_ecx, 13
bind_nec, 13
binomial, 28

brm, 15, 37
brmsfit, 10, 11, 28, 54, 64, 65
brmsformula, 71
brmsprior, 5–7, 15, 37, 73

call, 4, 6, 7, 9–11, 14, 15, 20, 21, 25, 27–29, 31, 33, 37, 47–49, 54, 56, 57, 59, 61, 62, 64, 68, 71, 73
call, 71
check_chains, 17
check_chains.bayesmanecfit, 18
check_chains.bayesnecfit, 18
check_chains.default, 19
check_data, 16, 20, 37
check_models, 21
check_priors, 21
check_priors.bayesmanecfit, 22
check_priors.bayesnecfit, 23
check_priors.default, 23

compare_endpoints, 24
compare_fitted, 25
compare_posterior, 26
customfamily, 11
data.frame, 10, 11, 13, 14, 20, 28, 37, 42–44, 65, 73
define_prior, 27
dispersion, 28
ecx, 29, 32, 42–44
ecx.bayesmanecfit, 30
ecx.bayesnecfit, 32
ecx.default, 33
expand_manec, 34
expand_nec, 35, 68
extract_simdat, 36

family, 14, 20, 21, 28, 37, 73
fit_bayesnec, 15, 20, 21, 36, 37

gaussian, 28
ggbnec, 38, 42–44
ggbnec.bayesmanecfit, 38, 39

ggbnec.bayesnecfit, 40
ggbnec.default, 41
ggbnec_data, 39–41, 42

ggbnec_data.bayesmanecfit, 42
ggbnec_data.bayesnecfit, 43
ggbnec_data.default, 44
ggplot, 38–41

integer, 70

is_bayesmanecfit, 45

is_bayesnecfit, 45

is_manecsummary, 45

is_necsummary, 46
INDEX

is_prebayesnecfit, 46
list, 4–8, 10, 11, 15, 20, 21, 24, 26–28, 35, 37, 47–50, 54, 64, 70, 73
log_lik_beta_binomial2, 47
logical, 9, 29, 31, 33, 45, 46, 49, 51, 52, 54, 56, 57, 59, 61, 62
loo_model_weights, 4, 6, 7, 15, 16, 35
make_good_inits, 47
make_inits, 47, 48, 48
manec_example, 49
manecsummary, 55, 67
manecsummary (manecsummary-class), 48
manecsummary-class, 48
models, 16, 50, 50
nec, 51, 53
nec.bayesmanecfit, 52
nec.bayesnecfit, 53
nec.default, 53
nec_data, 49, 55
necsummary, 49, 67
necsummary (necsummary-class), 54
necsummary-class, 54
nsec, 55, 58
nsec.bayesmanecfit, 57
nsec.bayesnecfit, 58
nsec.default, 59
numeric, 12, 13, 28, 47, 61–63, 70
plot, 61, 62
plot.bayesmanecfit, 60
plot.bayesnecfit, 61
poisson, 28
posterior_epred_beta_binomial2, 63
posterior_predict_beta_binomial2, 63
prebayesnecfit (prebayesnecfit-class), 64
prebayesnecfit-class, 64
predict.bayesmanecfit, 64
predict.bayesnecfit, 65
prep_raw_data, 65
print.bayesmanecfit, 66
print.bayesnecfit, 66
print.manecsummary, 67
print.necsummary, 67
prior, 28
pull_out, 68
pull_prior, 68
rhat.bayesmanecfit, 69
rhat.bayesnecfit, 69
sample_priors, 70
set_distribution, 70
show_params, 16, 50, 71
summary.bayesmanecfit, 67, 72
summary.bayesnecfit, 67, 72
validate_family, 73
validate_priors, 73