Package 'OncoBayes2'

August 4, 2026

Type Package

Title Bayesian Logistic Regression for Oncology Dose-Escalation Trials

Description Bayesian logistic regression model with optional EXchangeability-NonEXchangeability parameter modelling for flexible borrowing from historical or concurrent data-sources. The safety model can guide dose-escalation decisions for adaptive oncology Phase I dose-escalation trials which involve an arbitrary number of drugs. Please refer to Neuenschwander et al. (2008) <doi:10.1002/sim.3230> and Neuenschwander et al. (2016) <doi:10.1080/19466315.2016.1174149> for details on the methodology.

Version 0.10-0

Depends R (>= 3.5.0), RBesT (>= 1.8-2)

Imports methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.32.0), rstantools (>= 2.4.0), posterior (>= 1.5.0), assertthat (>= 0.2.1), checkmate, Formula, bayesplot (>= 1.4.0), dplyr (>= 1.1.0), tibble, tidyr (>= 1.0.0), abind, rlang (>= 0.3.0), tidyselect, utils, stats, matrixStats, lifecycle, grDevices, Rdpack

LinkingTo BH (>= 1.75.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.32.0), StanHeaders (>= 2.32.0)

License GPL (>= 3)

LazyData true

NeedsCompilation yes

UseLTO true

URL https://opensource.nibr.com/OncoBayes2/

BugReports https://github.com/Novartis/OncoBayes2/issues

Suggests rmarkdown, knitr, testthat (>= 2.0.0), withr, mvtnorm, vdiffr, cmdstanr (>= 0.4.0), ggplot2 (>= 3.0.0), scales, ragg

Additional_repositories https://stan-dev.r-universe.dev

VignetteBuilder knitr

RdMacros Rdpack

Biarch true

SystemRequirements GNU make, pandoc (>= 1.12.3), pngquant, C++17

Encoding UTF-8

Config/roxygen2/version 8.0.0

Author Novartis Pharma AG [cph],
Sebastian Weber [aut, cre],
Lukas A. Widmer [aut],
Andrew Bean [aut],
Trustees of Columbia University [cph] (R/stanmodels.R, configure, configure.win)

Maintainer Sebastian Weber <sebastian.weber@novartis.com>

Repository CRAN

Date/Publication 2026-08-04 21:30:02 UTC

Contents

bind_rows_0 . . . 3
blrm_exnex . . . 4
blrm_formula_linear . . . 12
blrm_formula_saturating . . . 13
blrm_trial . . . 14
codata_combo2 . . . 18
critical_quantile . . . 19
diagnostic-quantities . . . 22
dose_info_combo2 . . . 24
draws-OncoBayes2 . . . 24
drug_info_combo2 . . . 26
example-combo2 . . . 27
example-combo2_trial . . . 29
example-combo3 . . . 31
example-single-agent . . . 34
example_model . . . 35
hist_combo2 . . . 36
hist_combo3 . . . 37
hist_SA . . . 38
lodds . . . 40
nsamples.blrmfit . . . 41
OncoBayes2 . . . 41
plot_blrm . . . 43
posterior_interval.blrmfit . . . 48
posterior_linpred.blrmfit . . . 49
posterior_predict.blrmfit . . . 50
predictive_interval.blrmfit . . . 52

prior_summary.blrmfit 53
summary.blrmfit 54
summary.blrm_trial 55
update.blrmfit 58
update.blrm_trial 59

Index **60**

`bind_rows_0` *Bind rows of multiple data frames with zero fill*

Description

A version of bind_rows out of dplyr that fills non-common columns with zeroes instead of NA. Gives an error if any of the input data contains NA already.

Usage

```
bind_rows_0(...)
```

Arguments

`...`	Data frames to combine, passed into bind_rows (see dplyr documentation)

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

library(tibble)

dose_info_A <- tibble(
  group_id = "hist_A",
  drug_A = 1
)

dose_info_B <- tibble(
  group_id = "hist_B",
  drug_B = 100 * (1:2)
)

bind_rows_0(dose_info_A, dose_info_B)

## Recover user set sampling defaults
options(.user_mc_options)
```

blrm_exnex *Bayesian Logistic Regression Model for N-compounds with EXNEX*

Description

Bayesian Logistic Regression Model (BLRM) for N compounds using EXchangability and NonEXchangability (EXNEX) modeling.

Usage

```
blrm_exnex(
  formula,
  data,
  prior_EX_mu_comp,
  prior_EX_mu_mean_comp,
  prior_EX_mu_sd_comp,
  prior_EX_tau_comp,
  prior_EX_tau_mean_comp,
  prior_EX_tau_sd_comp,
  prior_EX_corr_eta_comp,
  prior_EX_mu_inter,
  prior_EX_mu_mean_inter,
  prior_EX_mu_sd_inter,
  prior_EX_tau_inter,
  prior_EX_tau_mean_inter,
  prior_EX_tau_sd_inter,
  prior_EX_corr_eta_inter,
  prior_is_EXNEX_inter,
  prior_is_EXNEX_comp,
  prior_EX_prob_comp,
  prior_EX_prob_inter,
  prior_NEX_mu_comp,
  prior_NEX_mu_mean_comp,
  prior_NEX_mu_sd_comp,
  prior_NEX_mu_inter,
  prior_NEX_mu_mean_inter,
  prior_NEX_mu_sd_inter,
  prior_tau_dist,
  sample_map = FALSE,
  iter = getOption("OncoBayes2.MC.iter", 2000),
  warmup = getOption("OncoBayes2.MC.warmup", 1000),
  save_warmup = getOption("OncoBayes2.MC.save_warmup", TRUE),
  thin = getOption("OncoBayes2.MC.thin", 1),
  init = getOption("OncoBayes2.MC.init", 0.5),
  chains = getOption("OncoBayes2.MC.chains", 4),
  cores = getOption("mc.cores", 1L),
  control = getOption("OncoBayes2.MC.control", list()),
```

```
  backend = getOption("OncoBayes2.MC.backend", "rstan"),
  prior_PD = FALSE,
  verbose = FALSE
)

## S3 method for class 'blrmfit'
print(x, ..., prob = 0.95, digits = 2)
```

Arguments

`formula` the model formula describing the linear predictors of the model. The lhs of the formula is a two-column matrix which are the number of occured events and the number of times no event occured. The rhs of the formula defines the linear predictors for the marginal models for each drug component, then the interaction model and at last the grouping and optional stratum factors of the models. These elements of the formula are separated by a vertical bar. The marginal models must follow a intercept and slope form while the interaction model must not include an interaction term. See the examples below for an example instantiation.

`data` optional data frame containing the variables of the model. If not found in `data`, the variables are taken from `environment(formula)`.

`prior_EX_mu_comp`
List of bivariate normal mixture priors for intercept and slope parameters $\boldsymbol{\mu}_i = (\mu_{\alpha i}, \mu_{\beta i})$ of each component. In case of a single drug model, then a mixture prior is accepted as well.

`prior_EX_mu_mean_comp`, `prior_EX_mu_sd_comp`
[Deprecated] Please use `prior_EX_mu_comp` instead. Mean and sd for the prior on the mean parameters $\boldsymbol{\mu}_i = (\mu_{\alpha i}, \mu_{\beta i})$ of each component. Two column matrix (intercept, log-slope) with one row per component.

`prior_EX_tau_comp`
List of bivariate normal mixture priors for heterogeniety parameter $\boldsymbol{\tau}_{si} = (\tau_{\alpha si}, \tau_{\beta si})$ of each stratum. If no differential discounting is required (i.e. if there is only one stratum $s = 1$), then it suffices to provide a bivariate normal mixture prior instead of a list with just one element.

`prior_EX_tau_mean_comp`, `prior_EX_tau_sd_comp`
[Deprecated] Please use `prior_EX_tau_comp` instead. Prior mean and sd for heterogeniety parameter $\boldsymbol{\tau}_{si} = (\tau_{\alpha si}, \tau_{\beta si})$ of each stratum. If no differential discounting is required (i.e. if there is only one stratum $s = 1$), then it is a two-column matrix (intercept, log-slope) with one row per component. Otherwise it is a three-dimensional array whose first dimension indexes the strata, second dimension indexes the components, and third dimension of length two for (intercept, log-slope).

`prior_EX_corr_eta_comp`
Prior LKJ correlation parameter for each component given as numeric vector. If missing, then a 1 is assumed corresponding to a marginal uniform prior of the correlation.

`prior_EX_mu_inter`
: Multivariate normal mixture prior for interaction parameter vector $\boldsymbol{\mu_\eta}$. Dimension must correspond to the number of interactions.

`prior_EX_mu_mean_inter`, `prior_EX_mu_sd_inter`
: **[Deprecated]** Please use `prior_EX_mu_inter` instead. Prior mean and sd for population mean parameters $\mu_{\eta k}$ of each interaction parameter. Vector of length equal to the number of interactions.

`prior_EX_tau_inter`
: List of multivariate normal mixture priors for heterogeniety interaction parameter vector $\boldsymbol{\tau_{\eta s}}$ of each stratum. If no differential discounting is required (i.e. if there is only one stratum $s = 1$), then it suffices to provide a mixture prior instead of a list with just one element.

`prior_EX_tau_mean_inter`, `prior_EX_tau_sd_inter`
: **[Deprecated]** Please use `prior_EX_tau_inter` instead. Prior mean and sd for heterogeniety parameter $\tau_{\eta sk}$ of each stratum. Matrix with one column per interaction and one row per stratum.

`prior_EX_corr_eta_inter`
: Prior LKJ correlation parameter for interaction given as numeric. If missing, then a 1 is assumed corresponding to a marginal uniform prior of the correlations.

`prior_is_EXNEX_inter`
: Defines if non-exchangability is admitted for a given interaction parameter. Logical vector of length equal to the number of interactions. If missing `FALSE` is assumed for all interactions.

`prior_is_EXNEX_comp`
: Defines if non-exchangability is admitted for a given component. Logical vector of length equal to the number of components. If missing `TRUE` is assumed for all components.

`prior_EX_prob_comp`
: Prior probability p_{ij} for exchangability of each component per group. Matrix with one column per component and one row per group. Values must lie in $[0 - 1]$ range.

`prior_EX_prob_inter`
: Prior probability $p_{\eta kj}$ for exchangability of each interaction per group. Matrix with one column per interaction and one row per group. Values must lie in $[0-1]$ range.

`prior_NEX_mu_comp`
: List of bivariate normal mixture priors $\boldsymbol{m}_{ij}$ and $\boldsymbol{s}_{ij} = \text{diag}(\boldsymbol{S}_{ij})$ of each component for non-exchangable case. If missing set to the same prior as given for the EX part. It is required that the specification be the same across groups j.

`prior_NEX_mu_mean_comp`, `prior_NEX_mu_sd_comp`
: **[Deprecated]** Please use `prior_NEX_mu_comp` instead. Prior mean $\boldsymbol{m}_{ij}$ and sd $\boldsymbol{s}_{ij} = \text{diag}(\boldsymbol{S}_{ij})$ of each component for non-exchangable case. Two column matrix (intercept, log-slope) with one row per component. If missing set to the same prior as given for the EX part. It is required that the specification be the same across groups j.

`prior_NEX_mu_inter`
: Multivariate normal mixture prior (mean $m_{\eta kj}$, sd $s_{\eta kj}$ and covariance) for the interaction parameter vector for non-exchangable case. Dimension must correspond to the number of interactions. If missing set to the same prior as given for the EX part.

`prior_NEX_mu_mean_inter`, `prior_NEX_mu_sd_inter`
: **[Deprecated]** Please use `prior_NEX_mu_inter` instead. Prior mean $m_{\eta kj}$ and sd $s_{\eta kj}$ for each interaction parameter for non-exchangable case. Vector of length equal to the number of interactions. If missing set to the same prior as given for the EX part.

`prior_tau_dist`
: Defines the distribution used for heterogeniety parameters. Choices are 0=fixed to it's mean, 1=log-normal, 2=truncated normal or `NULL` shutting off the hierarchical structure of the model.

`sample_map`
: Logical flag (defaults to `FALSE`) controlling inclusion of MAP priors for each stratum defined as part of the generated posterior. If set to `TRUE` then the posterior samples will contain `map_log_beta` and `map_eta` variables.

`iter`
: number of iterations (including warmup).

`warmup`
: number of warmup iterations.

`save_warmup`
: save warmup samples (`TRUE` / `FALSE`). Only if set to `TRUE`, then all random variables are saved in the posterior. This substantially increases the storage needs of the posterior.

`thin`
: period of saving samples.

`init`
: positive number to specify uniform range on unconstrained space for random initialization. See `stan`.

`chains`
: number of Markov chains.

`cores`
: number of cores for parallel sampling of chains.

`control`
: additional sampler parameters for NuTS algorithm.

`backend`
: sets Stan backend to be used. Possible choices are `"rstan"` (default) or `"cmdstanr"`.

`prior_PD`
: Logical flag (defaults to `FALSE`) indicating if to sample the prior predictive distribution instead of conditioning on the data.

`verbose`
: Logical flag (defaults to `FALSE`) controlling if additional output like stan progress is reported.

`x`
: `blrmfit` object to print

`...`
: not used in this function

`prob`
: central probability mass to report, i.e. the quantiles 0.5-prob/2 and 0.5+prob/2 are displayed. Multiple central widths can be specified.

`digits`
: number of digits to show

Details

`blrm_exnex` is a flexible function for Bayesian meta-analytic modeling of binomial count data. In particular, it is designed to model counts of the number of observed dose limiting toxicities (DLTs) by dose, for guiding dose-escalation studies in Oncology. To accommodate dose escalation over

more than one agent, the dose may consist of combinations of study drugs, with any number of treatment components.

In the simplest case, the aim is to model the probability π that a patient experiences a DLT, by complementing the binomial likelihood with a monotone logistic regression

$$\text{logit}\,\pi(d) = \log\,\alpha + \beta\,t(d),$$

where $\beta > 0$. Most typically, d represents the dose, and $t(d)$ is an appropriate transformation, such as $t(d) = \log(d/d^*)$. A joint prior on $\boldsymbol{\theta} = (\log\,\alpha, \log\,\beta)$ completes the model and ensures monotonicity $\beta > 0$.

Many extensions are possible. The function supports general combination regimens, and also provides framework for Bayesian meta-analysis of dose-toxicity data from multiple historical and concurrent sources.

For an example of a single-agent trial refer to `example-single-agent()`.

Value

The function returns a S3 object of type `blrmfit`.

Functions

- `print(blrmfit)`: print function.

Combination of two treatments

For a combination of two treatment components, the basic modeling framework is that the DLT rate $\pi(d_1, d_2)$ is comprised of (1) a "no-interaction" baseline model $\tilde{\pi}(d_1, d_2)$ driven by the single-agent toxicity of each component, and (2) optional interaction terms $\gamma(d_1, d_2)$ representing synergy or antagonism between the drugs. On the log-odds scale,

$$\text{logit}\,\pi(d_1, d_2) = \text{logit}\,\tilde{\pi}(d_1, d_2) + \eta\,\gamma(d_1, d_2).$$

The "no interaction" part $\tilde{\pi}(d_1, d_2)$ represents the probability of a DLT triggered by either treatment component acting *independently*. That is,

$$\tilde{\pi}(d_1, d_2) = 1 - (1 - \pi_1(d_1))(1 - \pi_2(d_2)).$$

In simple terms, P(no DLT for combination) = P(no DLT for drug 1) * P(no DLT from drug 2). To complete this part, the treatment components can then be modeled with monotone logistic regressions as before.

$$\text{logit}\,\pi_i(d_i) = \log\,\alpha_i + \beta_i\,t(d_i),$$

where $t(d_i)$ is a monotone transformation of the doses of the respective drug component i, such as $t(d_i) = \log(d_i/d_i^*)$.

The inclusion of an interaction term $\gamma(d_1, d_2)$ allows DLT rates above or below the "no-interaction" rate. The magnitude of the interaction term may also be made dependent on the doses (or other covariates) through regression. As an example, one could let

$$\gamma(d_1, d_2) = \frac{d_1}{d_1^*}\frac{d_1}{d_2^*}.$$

The specific functional form is specified in the usual notation for a design matrix. The interaction model must respect the constraint that whenever any dose approaches zero, then the interaction term must vanish as well. Therefore, the interaction model must not include an intercept term which would violate this consistency requirement. A dual combination example can be found in `example-combo2()`.

General combinations

The model is extended to general combination treatments consisting of N components by expressing the probability π on the logit scale as

$$\text{logit}\,\pi(d_1, \ldots, d_N) = \text{logit}\Big(1 - \prod_{i=1}^{N}(1 - \pi_i(d_i))\Big) + \sum_{k=1}^{K}\eta_k\,\gamma_k(d_1, \ldots, d_N),$$

Multiple drug-drug interactions among the N components are now possible, and are represented through the K interaction terms γ_k.

Regression models can be again be specified for each π_i and γ_k, such as

$$\text{logit}\,\pi_i(d_i) = \log\,\alpha_i + \beta_i\,t(d_i)$$

Interactions for some subset $I(k) \subset \{1, \ldots, N\}$ of the treatment components can be modeled with regression as well, for example on products of doses,

$$\gamma_k(d_1, \ldots, d_N) = \prod_{i \in I(k)}\frac{d_i}{d_i^*}.$$

For example, $I(k) = \{1, 2, 3\}$ results in the three-way interaction term

$$\frac{d_1}{d_1^*}\frac{d_2}{d_2^*}\frac{d_3}{d_3^*}$$

for drugs 1, 2, and 3.

For a triple combination example please refer to `example-combo3()`.

Meta-analytic framework

Information on the toxicity of a drug may be available from multiple studies or sources. Furthermore, one may wish to stratify observations within a single study (for example into groups of patients corresponding to different geographic regions, or multiple dosing `dose_info` corresponding to different schedules).

`blrm_exnex` provides tools for robust Bayesian hierarchical modeling to jointly model data from multiple sources. An additional index $j = 1, \ldots, J$ on the parameters and observations denotes the J groups. The resulting model allows the DLT rate to differ across the groups. The general N-component model becomes

$$\text{logit}\,\pi_j(d_1,\ldots,d_N) = \text{logit}\Big(1-\prod_{i=1}^{N}(1-\pi_{ij}(d_i))\Big) + \sum_{k=1}^{K}\eta_{kj}\,\gamma_k(d_1,\ldots,d_N),$$

for groups $j = 1,\ldots,J$. The component toxicities π_{ij} and interaction terms γ_k are modelled, as before, through regression. For example, π_{ij} could be a logistic regression on $t(d_i) = \log(d_i/d_i^*)$ with intercept and log-slope $\boldsymbol{\theta}_{ij}$, and γ_k regressed with coefficient η_{kj} on a product $\prod_{i\in I(k)}(d_i/d_i^*)$ for some subset $I(k)$ of components.

Thus, for $j = 1,\ldots,J$, we now have group-specific parameters $\boldsymbol{\theta}_{ij} = (\log\alpha_{ij}, \log\beta_{ij})$ and $\boldsymbol{\nu}_j = (\eta_{1j},\ldots,\eta_{Kj})$ for each component $i = 1,\ldots,N$ and interaction $k = 1,\ldots,K$.

The structure of the prior on $(\boldsymbol{\theta}_{i1},\ldots,\boldsymbol{\theta}_{iJ})$ and $(\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_J)$ determines how much information will be shared across groups j. Several modeling choices are available in the function.

- *EX (Full exchangeability):* One can assume the parameters are conditionally exchangeable given hyperparameters

  $$\boldsymbol{\theta}_{ij} \sim \text{N}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}i}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}i}\big),$$

  independently across groups $j = 1,\ldots,J$ and treatment components $i = 1,\ldots,N$. The covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}i}$ captures the patterns of cross-group heterogeneity, and is parametrized with standard deviations $\boldsymbol{\tau}_{\boldsymbol{\theta}i} = (\tau_{\alpha i}, \tau_{\beta i})$ and the correlation ρ_i. Similarly for the interactions, the fully-exchangeable model is

  $$\boldsymbol{\nu}_j \sim \text{N}\big(\boldsymbol{\mu}_{\boldsymbol{\nu}}, \boldsymbol{\Sigma}_{\boldsymbol{\nu}}\big)$$

  for groups $j = 1,\ldots,J$ and interactions $k = 1,\ldots,K$, and the prior on the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\nu}}$ captures the amount of heterogeneity expected in the interaction terms a-priori. The covariance is again parametrized with standard deviations $(\tau_{\eta 1},\ldots,\tau_{\eta K})$ and its correlation matrix.
- *Differential discounting:* For one or more of the groups $j = 1,\ldots,J$, larger deviations of $\boldsymbol{\theta}_{ij}$ may be expected from the mean $\boldsymbol{\mu}_i$, or of the interactions η_{kj} from the mean $\mu_{\eta,k}$. Such differential heterogeneity can be modeled by mapping the groups $j = 1,\ldots,J$ to *strata* through $s_j \in \{1,\ldots,S\}$, and modifying the model specification to

  $$\boldsymbol{\theta}_{ij} \sim \text{N}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}i}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}ij}\big),$$

  where

  $$\boldsymbol{\Sigma}_{\boldsymbol{\theta}ij} = \begin{pmatrix} \tau^2_{\alpha s_j i} & \rho_i\tau_{\alpha s_j i}\tau_{\beta s_j i} \\ \rho_i\tau_{\alpha s_j i}\tau_{\beta s_j i} & \tau^2_{\beta s_j i} \end{pmatrix}.$$

  For the interactions, the model becomes

  $$\boldsymbol{\nu}_j \sim \text{N}\big(\boldsymbol{\mu}_{\boldsymbol{\nu}}, \boldsymbol{\Sigma}_{\boldsymbol{\nu}j}\big),$$

  where the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\nu}j}$ is modelled as stratum specific standard deviations $(\tau_{\eta 1 s_j},\ldots,\tau_{\eta K s_j})$ and a stratum independent correlation matrix. Each stratum $s = 1,\ldots,S$ then corresponds to its own set of standard deviations τ leading to different discounting per stratum. Independent priors are specified for the component parameters $\tau_{\alpha si}$ and $\tau_{\beta si}$ and for the interaction parameters $\tau_{\eta sk}$ for each stratum $s = 1,\ldots,S$. Inference for strata s where the prior is centered on larger values of the τ parameters will exhibit less shrinkage towards the the means, $\boldsymbol{\mu}_{\boldsymbol{\theta}i}$ and $\boldsymbol{\mu}_{\boldsymbol{\nu}}$ respectively.

- *EXNEX (Partial exchangeability):* Another mechansim for increasing robustness is to introduce mixture priors for the group-specific parameters, where one mixture component is shared across groups, and the other is group-specific. The result, known as an EXchangeable-NonEXchangeable (EXNEX) type prior, has a form

$$\boldsymbol{\theta}_{ij} \sim p_{\boldsymbol{\theta}ij}\,\mathrm{N}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}i}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}i}\big) + (1 - p_{\boldsymbol{\theta}ij})\,\mathrm{N}\big(\boldsymbol{m}_{\boldsymbol{\theta}ij}, \boldsymbol{S}_{\boldsymbol{\theta}ij}\big)$$

  when applied to the treatment-component parameters, and

$$\boldsymbol{\nu}_{kj} \sim p_{\boldsymbol{\nu}_{kj}}\,\mathrm{N}\big(\mu_{\boldsymbol{\nu}}, \boldsymbol{\Sigma}_{\boldsymbol{\nu}}\big)_k + (1 - p_{\boldsymbol{\nu}_{kj}})\,\mathrm{N}(m_{\boldsymbol{\nu}_{kj}}, s^2_{\boldsymbol{\nu}_{kj}})$$

  when applied to the interaction parameters. The *exchangeability weights* $p_{\boldsymbol{\theta}ij}$ and $p_{\boldsymbol{\nu}_{kj}}$ are fixed constants in the interval $[0, 1]$ that control the degree to which inference for group j is informed by the exchangeable mixture components. Larger values for the weights correspond to greater exchange of information, while smaller values increase robustness in case of outlying observations in individual groups j.

References

Neuenschwander B, Roychoudhury S, Schmidli H (2016). "On the use of co-data in clinical trials." *Statistics in Biopharmaceutical Research*, **8**(3), 345–354. doi:10.1080/19466315.2016.1174149.

Neuenschwander B, Wandel S, Roychoudhury S, Bailey S (2016). "Robust exchangeability designs for early phase clinical trials with multiple strata." *Pharmaceutical Statistics*, **15**(2), 123–134. doi:10.1002/pst.1730.

Neuenschwander B, Branson M, Gsponer T (2008). "Critical aspects of the Bayesian approach to phase I cancer trials." *Statistics in Medicine*, **27**(13), 2420–2439. doi:10.1002/sim.3230.

Neuenschwander B, Matano A, Tang Z, Roychoudhury S, Wandel S, Bailey S (2014). "A Bayesian Industry Approach to Phase I Combination Trials in Oncology." In *Statistical Methods in Drug Combination Studies*, volume 69. CRC Press. doi:10.1201/b179659.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

# fit an example model. See documentation for "combo3" example
example_model("combo3")

# print a summary of the prior
prior_summary(blrmfit, digits = 3)

# print a summary of the posterior (model parameters)
print(blrmfit)

# summary of posterior for DLT rate by dose for observed covariate levels
summ <- summary(blrmfit, interval_prob = c(0, 0.16, 0.33, 1))
```

```
print(cbind(hist_combo3, summ))

# summary of posterior for DLT rate by dose for new set of covariate levels
newdata <- expand.grid(
  stratum_id = "BID", group_id = "Combo",
  drug_A = 400, drug_B = 800, drug_C = c(320, 400, 600, 800),
  stringsAsFactors = FALSE
)
summ_pred <- summary(blrmfit, newdata = newdata, interval_prob = c(0, 0.16, 0.33, 1))
print(cbind(newdata, summ_pred))

# update the model after observing additional data
newdata$num_patients <- rep(3, nrow(newdata))
newdata$num_toxicities <- c(0, 1, 2, 2)
library(dplyr)
blrmfit_new <- update(blrmfit,
  data = rbind(hist_combo3, newdata) %>%
    arrange(stratum_id, group_id)
)

# updated posterior summary
summ_upd <- summary(blrmfit_new, newdata = newdata, interval_prob = c(0, 0.16, 0.33, 1))
print(cbind(newdata, summ_upd))
## Recover user set sampling defaults
options(.user_mc_options)
```

`blrm_formula_linear` *Build a BLRM formula with linear interaction term in logit-space*

Description

`blrm_formula_linear` is a convenience function for generating a formula for `blrm_trial` and `blrm_exnex` with an interaction of the form:

$$\eta \prod_{i=1}^{N} (d_i / d_i^*))$$

Usage

```
blrm_formula_linear(
  ref_doses,
  max_interaction_level = 2,
  specific_interaction_terms = NULL
)
```

Arguments

`ref_doses`	Numeric vector of reference doses with names corresponding to drug names
`max_interaction_level`	Highest interaction order to consider $[1 - Inf]$. Default: 2
`specific_interaction_terms`	List of custom interaction terms to generate (e.g. list(c("drug1", "drug2"), c("drug1", "drug3"))). Default: NULL

Value

The function returns an object of class `blrm_formula`.

Examples

```
ref_doses <- c(drug_A = 10, drug_B = 20)

# can be used with blrm_trial
blrm_formula_linear(ref_doses)
```

`blrm_formula_saturating` *Build a BLRM formula with saturating interaction term in logit-space*

Description

`blrm_formula_saturating` is a convenience function for generating a formula for `blrm_trial` and `blrm_exnex` with an interaction of the form:

$$2\eta \frac{\prod_{i=1}^{N}(d_i/d_i^*)}{1+\prod_{i=1}^{N}(d_i/d_i^*)}$$

Usage

```
blrm_formula_saturating(
  ref_doses,
  max_interaction_level = 2,
  specific_interaction_terms = NULL
)
```

Arguments

`ref_doses`	Numeric vector of reference doses with names corresponding to drug names
`max_interaction_level`	Highest interaction order to consider $[1 - Inf]$. Default: 2
`specific_interaction_terms`	List of custom interaction terms to generate (e.g. `list(c("drug1", "drug2"), c("drug1", "drug3"))`).

Value

The function returns an object of class `blrm_formula`.

References

Widmer LA, Bean A, Ohlssen D, Weber S (2023). "Principled Drug-Drug Interaction Terms for Bayesian Logistic Regression Models of Drug Safety in Oncology Phase I Combination Trials." *arXiv pre-print*. doi:10.48550/arXiv.2302.11437.

Examples

```
ref_doses <- c(drug_A = 10, drug_B = 20)

# can be used with blrm_trial
blrm_formula_saturating(ref_doses)
```

`blrm_trial` *Dose-Escalation Trials guided by Bayesian Logistic Regression Model*

Description

`blrm_trial` facilitates the conduct of dose escalation studies guided by Bayesian Logistic Regression Models (BLRM). While the `blrm_exnex` only fits the BLRM model to data, the `blrm_trial` function standardizes the specification of the entire trial design and provides various standardized functions for trial data accrual and derivation of model summaries needed for dose-escalation decisions.

Usage

```
blrm_trial(
  data,
  dose_info,
  drug_info,
  simplified_prior = FALSE,
  EXNEX_comp = FALSE,
  EX_prob_comp_hist = 1,
  EX_prob_comp_new = 0.8,
  EXNEX_inter = FALSE,
  EX_prob_inter = 1,
  formula_generator = blrm_formula_saturating,
  interval_prob = c(0, 0.16, 0.33, 1),
 interval_max_mass = c(prob_underdose = 1, prob_target = 1, prob_overdose = 0.25),
  ...
)

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

Arguments

`data`	dose-toxicity data available at design stage of trial
`dose_info`	specificaion of the dose levels as planned for the ongoing trial arms.
`drug_info`	specification of drugs used in trial arms.
`simplified_prior`	logical (defaults to `FALSE`) indicating whether a simplified prior should be employed based on the `reference_p_dlt` values provided in `drug_info`. **Warning:** The simplified prior will change between releases. Please read instructions below in the respective section for the simplified prior.
`EXNEX_comp`	logical (default to `TRUE`) indicating whether EXchangeable-NonEXchangeable priors should be employed for all component parameters
`EX_prob_comp_hist`	prior weight ($[0, 1]$, default to 1) on exchangeability for the component parameters in groups representing historical data
`EX_prob_comp_new`	prior weight ($[0, 1]$, default to 0.8) on exchangeability for the component parameters in groups representing new or concurrent data
`EXNEX_inter`	logical (default to `FALSE`) indicating whether EXchangeable-NonEXchangeable priors should be employed for all interaction parameters
`EX_prob_inter`	prior weight ($[0, 1]$, defaults to 0.8) on exchangeability for the interaction parameters
`formula_generator`	formula generation function (see for example `blrm_formula_linear` or `blrm_formula_saturating`). The formula generator defines the employed interaction model.
`interval_prob`	defines the interval probabilities reported in the standard outputs. Defaults to `c(0, 0.16, 0.33, 1)`.
`interval_max_mass`	named vector defining for each interval of the `interval_prob` vector a maxmimal admissable probability mass for a given dose level. Whenever the posterior probability mass in a given interval exceeds the threshold, then the Escalation With Overdose Control (EWOC) criterion is considered to be not fullfilled. Dose levels not fullfilling EWOC are ineligible for the next cohort of patients. The default restricts the overdose probability to less than 0.25.
`...`	Additional arguments are forwarded to `blrm_exnex`, i.e. for the purpose of prior specification.
`x`	`blrm_trial` object to print

Details

`blrm_trial` constructs an object of class `blrm_trial` which stores the compelte information about the study design of a dose-escalation trial. The study design is defined through the data sets (see sections below for a definition of the columns):

data (historical data) The `data` argument defines available dose-toxicity data at the design stage of the trial. Together with the prior of model (without any data) this defines the prior used for the trial conduct.

dose_info Definition of the pre-specified dose levels explored in the ongoing trial arms. Thus, all dose-toxcitiy trial data added to the object is expected correspond to one of the dose levels in the pre-defined set of dose_info.

drug_info Determines the drugs used in the trial, their units, reference dose level and optionally defines the expected probability for a toxicity at the reference dose.

Once the `blrm_trial` object is setup the complete trial design is specified and the model is fitted to the given `data`. This allows evaluation of the pre-specified dose levels of the trial design wrt. to safety, i.e. whether the starting dose of the trial fullfills the escalate with overdose criterion (EWOC) condition.

The `blrm_trial` trial can also be constructed in a 2-step process which allows for a more convenient specification of the prior since meta data like number of drugs and the like can be used. See the example section for details.

After setup of the initial `blrm_trial` object additional data is added through the use of the `update` method which has a `add_data` argument intended to add data from the ongoing trial. The `summary` function finally allows to extract various model summaries. In particular, the EWOC criterion can be calculated for the pre-defined dose levels of the trial.

Value

The function returns an object of class `blrm_trial`.

Methods (by generic)

- `print(blrm_trial)`: print function.

Simplified prior

As a convenience for the user, a simplified prior can be specifed whenever the `reference_p_dlt` column is present in the `drug_info` data set. However, the user is **warned** that the simplified prior will change in future releases of the package and thus **we strongly discourage the use of the simplified prior for setting up trial designs**. The functionality is intended to provide the user a quick start and as a starting point. The actually instantiated prior can be seen as demonstrated below in the examples.

Input data

The data given to the `data` argument of `blrm_trial` is considered as the available at design stage of the trial. The collected input data thus does not necessarily need to have the same dose levels as the pre-specified dose_info for the ongoing trial(s). It's data columns must include, but are not limited to:

group_id study

stratum_id optional, only required for differential discounting of groups

num_patients number of patients

num_toxicities number of toxicities

drug_A Columns for the dose of each treatment component, with column names matching the `drug_name` values specified in the `drug_info` argument of `blrm_trial`

Drug info data

The drug information data-set defines drug properties. The fields included are:

drug_name name of drug which is also used as column name for the dose

dose_ref reference dose

dose_unit units used for drug amounts

reference_p_dlt optional; if provided, allows setup of a simplified prior

Dose info data

The `drug_info` data-set pre-specifies the dose levels of the ongoing trial. Thus, all data added to the `blrm_trial` through the `update` command must be consistent with the pre-defined dose levels as no other than those pre-specified ones can be explored in an ongoing trial.

dose_id optional column which assigns a unique id to each group_id/dose combination. If not specified the column is internally generated.

group_id study

drug_A Columns for the dose of each treatment component, with column names matching the `drug_name` values specified in the `drug_info` argument of `blrm_trial`

References

Babb J, Rogatko A, Zacks S (1998). "Cancer phase I clinical trials: efficient dose escalation with overdose control." *Statistics in Medicine*, **17**(10), 1103–1120. doi:10.1002/(SICI)10970258(19980530)17:10<1103::AID-SIM793>3.0.CO;29.

Neuenschwander B, Roychoudhury S, Schmidli H (2016). "On the use of co-data in clinical trials." *Statistics in Biopharmaceutical Research*, **8**(3), 345–354. doi:10.1080/19466315.2016.1174149.

See Also

Other blrm_trial combo2 example: `dose_info_combo2`, `drug_info_combo2`, `example-combo2_trial`

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)


# construct initial blrm_trial object from built-in example datasets
combo2_trial_setup <- blrm_trial(
  data = hist_combo2,
  dose_info = dose_info_combo2,
  drug_info = drug_info_combo2,
  simplified_prior = TRUE
)
```

```
# extract blrm_call to see setup of the prior as passed to blrm_exnex
summary(combo2_trial_setup, "blrm_exnex_call")

# Warning: The simplified prior will change between releases!
# please refer to the combo2_trial example for a complete
# example. You can obtain this example with
# ?example-combo2_trial
# or by running
# example_model("combo2_trial")

## Recover user set sampling defaults
options(.user_mc_options)
```

`codata_combo2` *Dataset: historical and concurrent data on a two-way combination*

Description

One of two datasets from the application described in Neuenschwander et al (2016). In the study `trial_AB`, the risk of DLT was studied as a function of dose for two drugs, drug A and drug B. Historical information on the toxicity profiles of these two drugs is available from single agent trials `trial_A` and `trial_B`. Another study `IIT` was run concurrently to `trial_AB`, and studies the same combination. A second dataset `hist_combo2` is available from this example, which includes only the data from the single agent studies, prior to the initiation of `trial_AB` and `IIT`.

Usage

```
codata_combo2
```

Format

A data frame with 20 rows and 5 variables:

group_id study

drug_A dose of Drug A

drug_B dose of Drug B

num_patients number of patients

num_toxicities number of DLTs

cohort_time cohort number of patients

References

Neuenschwander B, Roychoudhury S, Schmidli H (2016). "On the use of co-data in clinical trials." *Statistics in Biopharmaceutical Research*, **8**(3), 345–354. doi:10.1080/19466315.2016.1174149.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

dref <- c(6, 960)

num_comp   <- 2 # two investigational drugs
num_inter  <- 1 # one drug-drug interaction needs to be modeled
num_groups <- nlevels(codata_combo2$group_id) # no stratification needed
num_strata <- 1 # no stratification needed

blrmfit <- blrm_exnex(
  cbind(num_toxicities, num_patients - num_toxicities) ~
    1 + I(log(drug_A / dref[1])) |
      1 + I(log(drug_B / dref[2])) |
      0 + I(drug_A / dref[1] * drug_B / dref[2]) |
      group_id,
  data = codata_combo2,
  prior_EX_mu_comp  = list(mixmvnorm(c(1, logit(0.2), 0, diag(c(2^2, 1)))),
                           mixmvnorm(c(1, logit(0.2), 0, diag(c(2^2, 1))))),
  prior_EX_tau_comp = list(mixmvnorm(c(1,
                                       log(0.250), log(0.125),
                                       diag(c(log(4)/1.96, log(4)/1.96)^2))),
                           mixmvnorm(c(1,
                                       log(0.250), log(0.125),
                                       diag(c(log(4)/1.96, log(4)/1.96)^2)))),
  prior_EX_mu_inter = mixmvnorm(c(1, 0, 1.121^2)),
  prior_EX_tau_inter = mixmvnorm(c(1, log(0.125), (log(4) / 1.96)^2)),
  prior_is_EXNEX_comp = rep(FALSE, num_comp),
  prior_is_EXNEX_inter = rep(FALSE, num_inter),
  prior_EX_prob_comp = matrix(1, nrow = num_groups, ncol = num_comp),
  prior_EX_prob_inter = matrix(1, nrow = num_groups, ncol = num_inter),
  prior_tau_dist = 1
)
## Recover user set sampling defaults
options(.user_mc_options)
```

---

critical_quantile *Critical quantile*

Description

Calculates the critical value of a model covariate for which a fixed quantile of the crude rate is equal to the specified (tail or interval) probability. The specified quantile can relate to the posterior or the predictive distribution of the response rate.

Usage

```
critical_quantile(object, ...)

## S3 method for class 'blrmfit'
critical_quantile(
  object,
  newdata,
  x,
  p,
  qc,
  lower.tail,
  interval.x,
  extendInt.x = c("auto", "no", "yes", "downX", "upX"),
  log.x,
  predictive = FALSE,
  maxiter = 100,
  ...
)

## S3 method for class 'blrm_trial'
critical_quantile(
  object,
  newdata,
  x,
  p,
  qc,
  lower.tail,
  interval.x,
  extendInt.x = c("auto", "no", "yes", "downX", "upX"),
  log.x,
  predictive = FALSE,
  maxiter = 100,
  ...
)
```

Arguments

object	fitted model object
...	not used in this function
newdata	optional data frame specifying for what to predict; if missing, then the data of the input model object is used
x	character giving the covariate name which is being search for to meet the critical conditions. This also supports 'tidy' parameter selection by specifying x = vars(...), where ... is specified the same way as in dplyr::select() and similar functions. Examples of using x in this way can be found in the examples. For blrm_trial methods, it defaults to the first entry in summary(blrm_trial, "drug_info")$drug_name.

p	cumulative probability corresponding to the critical quantile range.
qc	if given as a sorted vector of length 2, then the two entries define the interval on the response rate scale which must correspond to the cumulative probability p. In this case lower.tail must be NULL or left missing. If qc is only a single number, then lower.tail must be set to complete the specification of the tail area.
lower.tail	defines if probabilities are lower or upper tail areas. Must be defined if qc is a single number and must not be defined if qc denotes an interval.
interval.x	interval the covariate x is searched. Defaults to the minimal and maximal value of x in the data set used. Whenever lower.tail is set such that it is known that a tail area is used, then the function will automatically attempt to enlarge the search interval since the direction of the inverted function is determined around the critical value.
extendInt.x	controls if the interval given in interval.x should be extended during the root finding. The default "auto" attempts to guess an adequate setting in cases thats possible. Other possible values are the same as for the extendInt argument of the stats::uniroot() function.
log.x	determines if during the numerical search for the critical value the covariate space is logarithmized. By default x is log transformed whenever all values are positive.
predictive	logical indicates if the critical quantile relates to the posterior (FALSE) or the predictive (TRUE) distribution. Defaults to FALSE.
maxiter	maximal number of iterations for root finding. Defaults to 100.

Details

The function searches for a critical value x_c of the covariate x (x) at which the integral over the model response in the specified interval π_1 to π_2 (qc) is equal to the given probability p (p). The given interval is considered a tail area when lower.tail is used such that $\pi_1 = 0$ for TRUE and $\pi_2 = 1$ for FALSE. At the critical value x_c the equality holds

$$p = \int_{\pi_1}^{\pi_2} p(\pi|x_c)\, d\pi.$$

Note that a solution not guranteed to exist under all circumstances. However, for a single agent model and when requesting a tail probability then a solution does exist. In case an interval probability is specified, then the solution may not necessarily exist and the root finding is confined to the range specified in interval.x.

In case the solution is requested for the predictive distribution (predictive=TRUE), then the respective problem solved leads to the equality of

$$p = \sum_{y=r_1=\lceil n\,\pi_1 \rceil}^{r_2=\lceil n\,\pi_2 \rceil} \int p(y|\pi, n)\, p(\pi|x_c)\, d\pi.$$

Furthermore, the covariate space is log transformed by default whenever all values of the covariate x are positive in the data set. Values of 0 are shifted into the positive domain by the machine precision to avoid issues with an ill-defined $\log(0)$.

For blrm_trial objects the default arguments for p, qc and lower.tail are set to correspond to the highest interval of interval_prob to be constrained by the respective interval_max_mass (which are defined as part of the blrm_trial objects). This will in the default case correspond to the EWOC metric. These defaults are only applied if p, qc and lower.tail are missing.

Value

Vector of length equal to the number of rows of the input data set with the crticial value for the covariate specified as x which fullfills the requirements as detailled above. May return NA for cases where no solution is found on the specified interval interval.x.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

# fit an example model. See documentation for "combo2" example
example_model("combo2", silent = TRUE)

# Find dose of drug_A at which EWOC criterium is just fulfilled
data_trial_ab <- subset(codata_combo2, group_id == "trial_AB")
drug_A_crit <- critical_quantile(blrmfit,
  newdata = data_trial_ab,
  x = "drug_A", p = 0.25, qc = 0.33,
  lower.tail = FALSE
)
data_trial_ab$drug_A <- drug_A_crit
summary(blrmfit, newdata = data_trial_ab, interval_prob = c(0, 0.16, 0.33, 1))

## Recover user set sampling defaults
options(.user_mc_options)
```

---

diagnostic-quantities *Extract Diagnostic Quantities of* **OncoBayes2** *Models*

Description

Extract quantities that can be used to diagnose sampling behavior of the algorithms applied by **Stan** at the back-end of **OncoBayes2**.

Usage

```
## S3 method for class 'blrmfit'
log_posterior(object, ...)
```

```
## S3 method for class 'blrmfit'
nuts_params(object, pars = NULL, ...)

## S3 method for class 'blrmfit'
rhat(object, pars = NULL, ...)

## S3 method for class 'blrmfit'
neff_ratio(object, pars = NULL, ...)
```

Arguments

object	A blrmfit or blrmtrial object.
...	Arguments passed to individual methods.
pars	An optional character vector of parameter names. For nuts_params these will be NUTS sampler parameter names rather than model parameters. If pars is omitted all parameters are included.

Details

For more details see bayesplot::bayesplot-extractors().

Value

The exact form of the output depends on the method.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

example_model("single_agent", silent = TRUE)

head(log_posterior(blrmfit))

np <- nuts_params(blrmfit)
str(np)
# extract the number of divergence transitions
sum(subset(np, Parameter == "divergent__")$Value)

head(rhat(blrmfit))
head(neff_ratio(blrmfit))

## Recover user set sampling defaults
options(.user_mc_options)
```

dose_info_combo2 *Dataset: trial dose information for a dual-agent combination study*

Description

The data set defines all pre-defined dose-levels which can be explored in the dual-agent example trial.

Usage

```
dose_info_combo2
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 42 rows and 4 columns.

Details

group_id study

drug_A drug A dose amount

drug_B drug B dose amount

dose_id unique id of record

See Also

Other blrm_trial combo2 example: `blrm_trial()`, `drug_info_combo2`, `example-combo2_trial`

draws-OncoBayes2 *Transform* `blrmfit` *or* `blrm_trial` *to* `draws` *objects*

Description

Transform a `blrmfit` or `blrm_trial` object to a format supported by the **posterior** package.

Usage

```
## S3 method for class 'blrmfit'
as_draws(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)

## S3 method for class 'blrmfit'
as_draws_matrix(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)

## S3 method for class 'blrmfit'
as_draws_array(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)
```

```
## S3 method for class 'blrmfit'
as_draws_df(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)

## S3 method for class 'blrmfit'
as_draws_list(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)

## S3 method for class 'blrmfit'
as_draws_rvars(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)

## S3 method for class 'blrm_trial'
as_draws(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)

## S3 method for class 'blrm_trial'
as_draws_matrix(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)

## S3 method for class 'blrm_trial'
as_draws_array(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)

## S3 method for class 'blrm_trial'
as_draws_df(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)

## S3 method for class 'blrm_trial'
as_draws_list(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)

## S3 method for class 'blrm_trial'
as_draws_rvars(x, variable = NULL, regex = FALSE, inc_warmup = FALSE, ...)
```

Arguments

x	A blrmfit or blrm_trial object.
variable	A character vector providing the variables to extract. By default, all variables are extracted.
regex	Logical; Should variable be treated as a (vector of) regular expressions? Any variable in x matching at least one of the regular expressions will be selected. Defaults to FALSE.
inc_warmup	Should warmup draws be included? Defaults to FALSE.
...	Arguments passed to individual methods (if applicable).

Details

To subset iterations, chains, or draws, use the posterior::subset_draws() method after transforming the input object to a draws object.

See Also

posterior::draws() posterior::subset_draws()

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

# fit an example model. See documentation for "combo2" example
example_model("combo2")

post <- as_draws(blrmfit)

## Recover user set sampling defaults
options(.user_mc_options)
```

---

drug_info_combo2 *Dataset: drug information for a dual-agent combination study*

Description

Data set describing the two drugs involved in the example for a dual-agent combination study.

Usage

```
drug_info_combo2
```

Format

A tibble with 2 rows (one per durg) and 4 columns:

drug_name name of drug

dose_ref reference dose

dose_unit units used for drug amounts

reference_p_dlt a-priori probability for a DLT at the reference dose

See Also

Other blrm_trial combo2 example: `blrm_trial()`, `dose_info_combo2`, `example-combo2_trial`

`example-combo2` *Two-drug combination example*

Description

Example using a combination of two experimental drugs.

Details

The following example is described in the reference Neuenschwander, B. et al (2016). The data are described in the help page for `codata_combo2`. In the study `trial_AB`, the risk of DLT was studied as a function of dose for two drugs, drug A and drug B. Historical information on the toxicity profiles of these two drugs was available from single agent trials `trial_A` and `trial_B`. Another study `IIT` was run concurrently to `trial_AB`, and studies the same combination.

The model described in Neuenschwander, et al (2016) is adapted as follows. For groups $j = 1, \ldots, 4$ representing each of the four sources of data mentioned above,

$$\text{logit}\, \pi_{1j}(d_1) = \log\, \alpha_{1j} + \beta_{1j}\, \log\Big(\frac{d_1}{d_1^*}\Big),$$

and

$$\text{logit}\, \pi_{2j}(d_2) = \log\, \alpha_{2j} + \beta_{2j}\, \log\Big(\frac{d_2}{d_2^*}\Big),$$

are logistic regressions for the single-agent toxicity of drugs A and B, respectively, when administered in group j. Conditional on the regression parameters $\boldsymbol{\theta}_{1j} = (\log\, \alpha_{1j}, \log\, \beta_{1j})$ and $\boldsymbol{\theta}_{2j} = (\log\, \alpha_{2j}, \log\, \beta_{2j})$, the toxicity $\pi_j(d_1, d_2)$ for the combination is modeled as the "no-interaction" DLT rate,

$$\tilde{\pi}_j(d_1, d_2) = 1 - (1 - \pi_{1j}(d_1))(1 - \pi_{2j}(d_2))$$

with a single interaction term added on the log odds scale,

$$\text{logit}\, \pi_j(d_1, d_2) = \text{logit}\, \tilde{\pi}_j(d_1, d_2) + \eta_j \frac{d_1}{d_1^*} \frac{d_2}{d_2^*}.$$

A hierarchical model across the four groups j allows dose-toxicity information to be shared through common hyperparameters.

For the component parameters $\boldsymbol{\theta}_{ij}$,

$$\boldsymbol{\theta}_{ij} \sim \text{BVN}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i).$$

For the mean, a further prior is specified as

$$\boldsymbol{\mu}_i = (\mu_{\alpha i}, \mu_{\beta i}) \sim \text{BVN}(\boldsymbol{m}_i, \boldsymbol{S}_i),$$

while in the manuscript the prior $\boldsymbol{m}_i = (\text{logit}\, 0.1, \log 1)$ and $\boldsymbol{S}_i = \text{diag}(3.33^2, 1^2)$ for each $i = 1, 2$ is used, we deviate here and use instead $\boldsymbol{m}_i = (\text{logit}\, 0.2, \log 1)$ and $\boldsymbol{S}_i = \text{diag}(2^2, 1^2)$. For the standard deviations and correlation parameters in the covariance matrix,

$$\boldsymbol{\Sigma}_i = \begin{pmatrix} \tau_{\alpha i}^2 & \rho_i \tau_{\alpha i} \tau_{\beta i} \\ \rho_i \tau_{\alpha i} \tau_{\beta i} & \tau_{\beta i}^2 \end{pmatrix},$$

the specified priors are $\tau_{\alpha i} \sim$ Log-Normal$(\log\ 0.25, ((\log 4)/1.96)^2)$,
$\tau_{\beta i} \sim$ Log-Normal$(\log\ 0.125, ((\log 4)/1.96)^2)$, and $\rho_i \sim \mathrm{U}(-1, 1)$ for $i = 1, 2$.

For the interaction parameters η_j in each group, the hierarchical model has

$$\eta_j \sim \mathrm{N}(\mu_\eta, \tau_\eta^2),$$

for $j = 1, \ldots, 4$, with $\mu_\eta \sim \mathrm{N}(0, 1.121^2)$ and $\tau_\eta \sim$ Log-Normal$(\log\ 0.125, ((\log 4)/1.96)^2)$.

Below is the syntax for specifying this fully exchangeable model in `blrm_exnex`.

References

Neuenschwander B, Roychoudhury S, Schmidli H (2016). "On the use of co-data in clinical trials." *Statistics in Biopharmaceutical Research*, **8**(3), 345–354. doi:10.1080/19466315.2016.1174149.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

dref <- c(6, 960)

num_comp   <- 2 # two investigational drugs
num_inter  <- 1 # one drug-drug interaction needs to be modeled
num_groups <- nlevels(codata_combo2$group_id) # no stratification needed
num_strata <- 1 # no stratification needed

blrmfit <- blrm_exnex(
  cbind(num_toxicities, num_patients - num_toxicities) ~
    1 + I(log(drug_A / dref[1])) |
      1 + I(log(drug_B / dref[2])) |
      0 + I(drug_A / dref[1] * drug_B / dref[2]) |
      group_id,
  data = codata_combo2,
  prior_EX_mu_comp  = list(mixmvnorm(c(1, logit(0.2), 0, diag(c(2^2, 1)))),
                           mixmvnorm(c(1, logit(0.2), 0, diag(c(2^2, 1))))),
  prior_EX_tau_comp = list(mixmvnorm(c(1,
                                       log(0.250), log(0.125),
                                       diag(c(log(4)/1.96, log(4)/1.96)^2))),
                           mixmvnorm(c(1,
                                       log(0.250), log(0.125),
                                       diag(c(log(4)/1.96, log(4)/1.96)^2)))),
  prior_EX_mu_inter = mixmvnorm(c(1, 0, 1.121^2)),
  prior_EX_tau_inter = mixmvnorm(c(1, log(0.125), (log(4) / 1.96)^2)),
  prior_is_EXNEX_comp = rep(FALSE, num_comp),
  prior_is_EXNEX_inter = rep(FALSE, num_inter),
  prior_EX_prob_comp = matrix(1, nrow = num_groups, ncol = num_comp),
  prior_EX_prob_inter = matrix(1, nrow = num_groups, ncol = num_inter),
```

```
  prior_tau_dist = 1
)
## Recover user set sampling defaults
options(.user_mc_options)
```

---

example-combo2_trial *Two-drug combination example using BLRM Trial*

Description

Example using `blrm_trial()` to guide the built-in two-drug combination study example.

Details

`blrm_trial()` is used to collect and store all relevant design information for the example. Subsequent use of the `update.blrm_trial()` command allows convenient model fitting via `blrm_exnex()`. The `summary.blrm_trial()` method allows exploration of the design and modeling results.

To run this example, use `example_model("combo2_trial")`. See `example_model()`.

See Also

Other blrm_trial combo2 example: `blrm_trial()`, `dose_info_combo2`, `drug_info_combo2`

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

library(tibble)
library(dplyr)
library(tidyr)

# Combo2 example using blrm_trial functionality

# construct initial blrm_trial object from built-in example datasets
combo2_trial_setup <- blrm_trial(
  data = hist_combo2,
  dose_info = dose_info_combo2,
  drug_info = drug_info_combo2,
  simplified_prior = FALSE
)

# summary of dimensionality of data structures
dims <- summary(combo2_trial_setup, "dimensionality")
```

```
# Fit the initial model with the historical data and fully specified prior


combo2_trial_start <- update(
   combo2_trial_setup,
   ## bivariate normal prior for drug A and drug B of intercept and
   ## log-slope
   prior_EX_mu_comp =
     replicate(2,
               mixmvnorm(c(1,
                           logit(0.2), 0,
                           diag(c(2^2, 1))))
             , FALSE),
   prior_EX_tau_comp =
     replicate(2,
               mixmvnorm(c(1,
                           log(0.25), log(0.125),
                           diag(c(log(4)/1.96, log(4)/1.96)^2)))
             , FALSE),
   prior_EX_mu_inter = mixmvnorm(c(1, 0, 1.121^2)),
   prior_EX_tau_inter = mixmvnorm(c(1, log(0.125), (log(4) / 1.96)^2)),
   prior_is_EXNEX_comp = c(FALSE, FALSE),
   prior_is_EXNEX_inter = FALSE,
   prior_EX_prob_comp = matrix(1,
     nrow = dims$num_groups,
     ncol = 2
   ),
   prior_EX_prob_inter = matrix(1,
     nrow = nlevels(dose_info_combo2$group_id),
     ncol = 1
   ),
   prior_tau_dist = 1
 )

# print summary of prior specification
prior_summary(combo2_trial_start)

# summarize inference at observed dose levels
summary(combo2_trial_start, "data_prediction")

# summarize inference at specified dose levels
summary(combo2_trial_start, "dose_prediction")


# Update again with new data

# using update() with data argument supplied
# dem <- update(combo2_trial_start, data = codata_combo2)

# alternate way using update() with add_data argument for
# new observations only (those collected after the trial
# design stage).
```

```
new_data <- filter(codata_combo2, cohort_time > 0)

combo2_trial <- update(combo2_trial_start, add_data = new_data)

summary(combo2_trial, "data") # cohort_time is tracked
summary(combo2_trial, "data_prediction")
summary(combo2_trial, "dose_prediction")

rm(dims, new_data)

## Recover user set sampling defaults
options(.user_mc_options)
```

`example-combo3` *Three-drug combination example*

Description

Example using a combination of two experimental drugs, with EXNEX and differential discounting.

Details

This dataset involves a hypothetical dose-escalation study of combination therapy with three treatment components. From two previous studies `HistAgent1` and `HistAgent2`, historical data is available on each of the treatments as single-agents, as well as two of the two-way combinations. However, due to a difference in treatment schedule between the `Combo` study and the historical studies, a stratification (through `stratum_id`) is made between the groups to allow differential discounting of the alternate-schedule data. The association is as below.

group_id (j):	stratum_id (s_j):
Combo (1)	BID (1)
HistAgent1 (2)	QD (2)
HistAgent2 (3)	QD (2)

For additional robustness, EXNEX priors are used for all group-level treatment components while not for the interaction parameters. This is to limit the amount of borrowing in case of significant heterogeneity across groups.

The complete model is as follows. As a function of doses d_1, d_2, d_3, the DLT rate in group j is, for $j = 1, \ldots, 3$,

$$\text{logit}\,\pi_j(d_1,d_2,d_3) = \text{logit}\Big(1-\prod_{i=1}^{3}(1-\pi_{ij}(d_i))\Big)+\eta_j^{(12)}\frac{d_1}{d_1^*}\frac{d_2}{d_2^*}+\eta_j^{(13)}\frac{d_1}{d_1^*}\frac{d_3}{d_3^*}+\eta_j^{(23)}\frac{d_2}{d_2^*}\frac{d_3}{d_3^*}+\eta_j^{(123)}\frac{d_1}{d_1^*}\frac{d_2}{d_2^*}\frac{d_3}{d_3^*}.$$

In group j each treatment component i toxicity is modeled with logistic regression,

$$\text{logit}\,\pi_{ij}(d_i) = \log\,\alpha_{ij} + \beta_{ij}\,\log\left(\frac{d_i}{d_i^*}\right).$$

The intercept and log-slope parameters $\boldsymbol{\theta}_{ij} = (\log\, \alpha_{ij}, \log\, \beta_{ij})$ are are given an EXNEX prior

$$\boldsymbol{\theta}_{ij} \sim p_{ij}\text{BVN}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_{ij}) + (1 - p_{ij})\text{BVN}(\boldsymbol{m}_{ij}, \boldsymbol{S}_{ij}),$$

where the exchangeability weights are all $p_{ij} = 0.9$. The NEX parameters are set to $\boldsymbol{m}_{ij} = (\text{logit}(1/3), \log\, 1)$, $\boldsymbol{S}_{ij} = \text{diag}(2^2, 1^2)$ for all components $i = 1, 2, 3$ and groups $j = 1, 2, 3$, and the EX parameters are modeled hierarchically. The mean of the exchangeable part has the distribution

$$\boldsymbol{\mu}_i = (\mu_{\alpha i}, \mu_{\beta i}) \sim \text{BVN}(\boldsymbol{m}_i, \boldsymbol{S}_i),$$

with $\boldsymbol{m}_i = (\text{logit}(1/3), \log 1)$ and $\boldsymbol{S}_i = \text{diag}(2^2, 1^2)$ for each component $i = 1, 2, 3$. For differentially discounting data from each schedule (QD and BID), the covariance parameters for the exchangeable part

$$\Sigma_{ij} = \begin{pmatrix} \tau^2_{\alpha s_j i} & \rho_i \tau_{\alpha s_j i} \tau_{\beta s_j i} \\ \rho_i \tau_{\alpha s_j i} \tau_{\beta s_j i} & \tau^2_{\beta s_j i} \end{pmatrix}.$$

are allowed to vary across groups j depending on their mapping to strata $s(j)$ as described above. For stratum $s = 1$ (`BID`, which contains only the group $j = 1$ (`Combo`)), the standard deviations are modeled as

$$\tau_{\alpha 1 i} \sim \text{Log-Normal}(\log\, 0.25, (\log 4/1.96)^2)$$

$$\tau_{\beta 1 i} \sim \text{Log-Normal}(\log\, 0.125, (\log 4/1.96)^2).$$

Whereas in stratum $s = 2$ (`QD`, which contains the historical groups $j = 2, 3$ (`HistData1`, `HistData2`)), the standard deviations are

$$\tau_{\alpha 2 i} \sim \text{Log-Normal}(\log\, 0.5, (\log 4/1.96)^2)$$

$$\tau_{\beta 2 i} \sim \text{Log-Normal}(\log\, 0.25, (\log 4/1.96)^2).$$

For all interaction parameters $\eta_j^{(12)}$, $\eta_j^{(13)}$, $\eta_j^{(23)}$, and $\eta_j^{(123)}$ ($j = 1, 2, 3$), the following prior is assumed:

$$\eta_j^{(\cdot)} \sim \text{N}(\mu_\eta^{(\cdot)}, {\tau_{\eta s_j}^{(\cdot)}}^2).$$

The exchangeability weights are $p_{\eta j}^{(\cdot)} = 0.9$ for all parameters with EXNEX. Here, for each $\mu_\eta^{(12)}$, $\mu_\eta^{(13)}$, $\mu_\eta^{(23)}$, and $\mu_\eta^{(123)}$, we take

$$\mu_\eta^{(\cdot)} \sim \text{N}(0, 1/2),$$

and for each $\tau_{\eta s}^{(12)}$, $\tau_{\eta s}^{(13)}$, $\tau_{\eta s}^{(23)}$, and $\tau_{\eta s}^{(123)}$,

$$\tau_{\eta s}^{(\cdot)} \sim \text{Log-Normal}(\log(0.25), (\log 2/1.96)^2),$$

for both strata $s = 1, 2$.

Below is the syntax for specifying this model in `blrm_exnex`.

References

Neuenschwander B, Roychoudhury S, Schmidli H (2016). "On the use of co-data in clinical trials." *Statistics in Biopharmaceutical Research*, **8**(3), 345–354. doi:10.1080/19466315.2016.1174149.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

## example combo3

library(abind)

dref <- c(500, 500, 1000)
num_comp <- 3
num_inter <- choose(3, 2) + 1
num_strata <- nlevels(hist_combo3$stratum_id)
num_groups <- nlevels(hist_combo3$group_id)

blrmfit <- blrm_exnex(
  cbind(num_toxicities, num_patients - num_toxicities) ~
    1 + I(log(drug_A / dref[1])) |
      1 + I(log(drug_B / dref[2])) |
      1 + I(log(drug_C / dref[3])) |
      0
      + I(drug_A / dref[1] * drug_B / dref[2])
        + I(drug_A / dref[1] * drug_C / dref[3])
        + I(drug_B / dref[2] * drug_C / dref[3])
        + I(drug_A / dref[1] * drug_B / dref[2] * drug_C / dref[3]) |
      stratum_id / group_id,
  data = hist_combo3,
 prior_EX_mu_comp = replicate(num_comp, mixmvnorm(c(1, logit(1/3), 0, diag(c(2^2, 1)))), FALSE),
  prior_EX_tau_comp = list(replicate(num_comp,
                                     mixmvnorm(c(1, log(c(0.25, 0.125)),
                                          diag(c(log(4)/1.96, log(4)/1.96)^2))), FALSE),
                           replicate(num_comp,
                                     mixmvnorm(c(1, log(2 * c(0.25, 0.125)),
                                         diag(c(log(4)/1.96, log(4)/1.96)^2))), FALSE)),
  prior_EX_mu_inter = mixmvnorm(c(1, rep.int(0, num_inter),
                                     diag((rep.int(sqrt(2) / 2, num_inter))^2))),
  prior_EX_tau_inter = replicate(num_strata,
                                 mixmvnorm(c(1, rep.int(log(0.25), num_inter),
                                   diag((rep.int(log(2) / 1.96, num_inter))^2))), FALSE),
  prior_EX_prob_comp = matrix(0.9, nrow = num_groups, ncol = num_comp),
  prior_EX_prob_inter = matrix(1.0, nrow = num_groups, ncol = num_inter),
  prior_is_EXNEX_comp = rep(TRUE, num_comp),
  prior_is_EXNEX_inter = rep(FALSE, num_inter),
  prior_tau_dist = 1,
  prior_PD = FALSE
)
## Recover user set sampling defaults
options(.user_mc_options)
```

`example-single-agent` *Single Agent Example*

Description

Example using a single experimental drug.

Details

The single agent example is described in the reference Neuenschwander, B. et al (2008). The data are described in the help page for `hist_SA`. In this case, the data come from only one study, with the treatment being only single agent. Hence the model specified does not involve a hierarchical prior for the intercept and log-slope parameters. The model described in Neuenschwander, et al (2008) is adapted as follows:

$$\text{logit}\,\pi(d) = \log\,\alpha + \beta\,\log\left(\frac{d}{d^*}\right),$$

where $d^* = 250$, and the prior for $\boldsymbol{\theta} = (\log\,\alpha, \log\,\beta)$ is

$$\boldsymbol{\theta} \sim \text{N}(\boldsymbol{m}, \boldsymbol{S}),$$

and $\boldsymbol{m} = (\text{logit}\,0.5, \log\,1)$ and $\boldsymbol{S} = \text{diag}(2^2, 1^2)$ are constants.

The above model is non-hierarchical. To disable the hierarchical model structure of the `blrm_exnex` framework, the user can specify the option `prior_tau_dist=NULL`. This will internally set all the heterogeneity parameters (τ_α^2 and τ_β^2) to zero.

References

Neuenschwander B, Branson M, Gsponer T (2008). "Critical aspects of the Bayesian approach to phase I cancer trials." *Statistics in Medicine*, **27**(13), 2420–2439. doi:10.1002/sim.3230.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

## Example from Neuenschwander, B., et al. (2009). Stats in Medicine

dref <- 50

## Since there is no prior information the hierarchical model
## is not used in this example by setting tau to (almost) 0.
blrmfit <- blrm_exnex(
  cbind(num_toxicities, num_patients - num_toxicities) ~
      1 + log(drug_A / dref) |
```

```
    0 |
    group_id,
  data = hist_SA,
  prior_EX_mu_comp = mixmvnorm(c(1, logit(1 / 2), log(1), diag(c(2^2, 1)))),
  ## Setting prior_tau_dist=NULL disables the hierarchical prior which is
  ## not required in this example as we analyze a single trial.
  prior_tau_dist = NULL,
  prior_PD = FALSE
)
## Recover user set sampling defaults
options(.user_mc_options)
```

`example_model` *Runs example models*

Description

Runs example models

Usage

```
example_model(topic, envir = parent.frame(), silent = FALSE)
```

Arguments

`topic`	example to run
`envir`	environment which the example is loaded into. Defaults to the caller environment.
`silent`	logical controlling if execution is run silently (defaults to `FALSE`)

Value

When topic is not specified a list of all possible topics is return. Whenever a valid topic is specified, the function inserts the example into the environment given and returns (invisibly) the updated environment.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)


## get a list of available examples
example_model()
```

```
## run 3 component example
example_model("combo3")

## Recover user set sampling defaults
options(.user_mc_options)
```

`hist_combo2` *Dataset: historical data on two single-agents to inform a combination study*

Description

One of two datasets from the application described in Neuenschwander et al (2016). The risk of DLT is to be studied as a function of dose for two drugs, drug A and drug B. Historical information on the toxicity profiles of these two drugs is available from single agent trials `trial_A` and `trial_B`. A second dataset `codata_combo2` is available from this application, which includes additional dose-toxicity data from `trial_AB` and `IIT` of the combination of Drugs A and B.

Usage

```
hist_combo2
```

Format

A tibble with 11 rows and 5 variables:

group_id study

drug_A dose of Drug A

drug_B dose of Drug B

num_patients number of patients

num_toxicities number of DLTs

cohort_time cohort number of patients

References

Neuenschwander B, Roychoudhury S, Schmidli H (2016). "On the use of co-data in clinical trials." *Statistics in Biopharmaceutical Research*, **8**(3), 345–354. doi:10.1080/19466315.2016.1174149.

`hist_combo3` *Dataset: historical and concurrent data on a three-way combination*

Description

This dataset involves a hypothetical dose-escalation study of combination therapy with three treatment components. From two previous studies `HistAgent1` and `HistAgent2`, historical data is available on each of the treatments as single-agents, as well as two of the two-way combinations. However, due to a difference in treatment schedule between the `Combo` study and the historical studies, a stratification (through `stratum_id`) is made between the groups to allow differential discounting of the alternate-dose data.

Usage

```
hist_combo3
```

Format

A data frame with 18 rows and 7 variables:

group_id study

drug_A dose of Drug A

drug_B dose of Drug B

drug_C dose of Drug C

num_patients number of patients

num_toxicities number of DLTs

stratum_id stratum for `group_id`'s used for differential discounting

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

## example combo3

library(abind)

dref <- c(500, 500, 1000)
num_comp <- 3
num_inter <- choose(3, 2) + 1
num_strata <- nlevels(hist_combo3$stratum_id)
num_groups <- nlevels(hist_combo3$group_id)
```

```
blrmfit <- blrm_exnex(
  cbind(num_toxicities, num_patients - num_toxicities) ~
    1 + I(log(drug_A / dref[1])) |
      1 + I(log(drug_B / dref[2])) |
      1 + I(log(drug_C / dref[3])) |
      0
      + I(drug_A / dref[1] * drug_B / dref[2])
        + I(drug_A / dref[1] * drug_C / dref[3])
        + I(drug_B / dref[2] * drug_C / dref[3])
        + I(drug_A / dref[1] * drug_B / dref[2] * drug_C / dref[3]) |
      stratum_id / group_id,
  data = hist_combo3,
 prior_EX_mu_comp = replicate(num_comp, mixmvnorm(c(1, logit(1/3), 0, diag(c(2^2, 1)))), FALSE),
  prior_EX_tau_comp = list(replicate(num_comp,
                                     mixmvnorm(c(1, log(c(0.25, 0.125)),
                                          diag(c(log(4)/1.96, log(4)/1.96)^2))), FALSE),
                           replicate(num_comp,
                                     mixmvnorm(c(1, log(2 * c(0.25, 0.125)),
                                         diag(c(log(4)/1.96, log(4)/1.96)^2))), FALSE)),
  prior_EX_mu_inter = mixmvnorm(c(1, rep.int(0, num_inter),
                                     diag((rep.int(sqrt(2) / 2, num_inter))^2))),
  prior_EX_tau_inter = replicate(num_strata,
                                 mixmvnorm(c(1, rep.int(log(0.25), num_inter),
                                   diag((rep.int(log(2) / 1.96, num_inter))^2))), FALSE),
  prior_EX_prob_comp = matrix(0.9, nrow = num_groups, ncol = num_comp),
  prior_EX_prob_inter = matrix(1.0, nrow = num_groups, ncol = num_inter),
  prior_is_EXNEX_comp = rep(TRUE, num_comp),
  prior_is_EXNEX_inter = rep(FALSE, num_inter),
  prior_tau_dist = 1,
  prior_PD = FALSE
)
## Recover user set sampling defaults
options(.user_mc_options)
```

`hist_SA` *Single-agent example*

Description

Example data from the application in Neuenschwander, et. al. 2008, from an "open-label, multicenter, non-comparative, dose-escalation cancer trial to characterize the safety, tolerability, and pharmacokinetic profile of a drug and to determine its MTD."

Usage

```
hist_SA
```

Format

A data frame with 5 rows and 4 variables:

group_id study

drug_A dose

num_patients number of patients

num_toxicities number of events

References

Neuenschwander B, Branson M, Gsponer T (2008). "Critical aspects of the Bayesian approach to phase I cancer trials." *Statistics in Medicine*, **27**(13), 2420–2439. doi:10.1002/sim.3230.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

## Example from Neuenschwander, B., et al. (2009). Stats in Medicine

dref <- 50

## Since there is no prior information the hierarchical model
## is not used in this example by setting tau to (almost) 0.
blrmfit <- blrm_exnex(
  cbind(num_toxicities, num_patients - num_toxicities) ~
      1 + log(drug_A / dref) |
      0 |
      group_id,
  data = hist_SA,
  prior_EX_mu_comp = mixmvnorm(c(1, logit(1 / 2), log(1), diag(c(2^2, 1)))),
  ## Setting prior_tau_dist=NULL disables the hierarchical prior which is
  ## not required in this example as we analyze a single trial.
  prior_tau_dist = NULL,
  prior_PD = FALSE
)
## Recover user set sampling defaults
options(.user_mc_options)
```

`lodds` *Logit (log-odds) and inverse-logit function.*

Description

Calculates the logit (log-odds) and inverse-logit.

Usage

```
logit(mu)

inv_logit(eta)
```

Arguments

`mu`	A numeric object with probabilies, with values in the in the range $[0, 1]$. Missing values (`NA`s) are allowed.
`eta`	A numeric object with log-odds values, with values in the range $[-\infty, \infty]$. Missing values (`NA`s) are allowed.

Details

Values of mu equal to 0 or 1 will return $-\infty$ or ∞ respectively.

Value

A numeric object of the same type as mu and eta containing the logits or inverse logit of the input values. The logit and inverse transformation equates to

$$\text{logit}(\mu) = \log(\mu/(1-\mu))$$

$$\text{logit}^{-1}(\eta) = \exp(\eta)/(1+\exp(\eta)).$$

Examples

```
logit(0.2)
inv_logit(-1.386)
```

nsamples.blrmfit *Return the number of posterior samples*

Description

Return the number of posterior samples

Usage

```
## S3 method for class 'blrmfit'
nsamples(object, ...)
```

Arguments

object	fitted model object
...	not used in this function

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)


## run single-agent analysis which defines blrmfit model object
example_model("single_agent", silent = TRUE)

nsamples(blrmfit)

## Recover user set sampling defaults
options(.user_mc_options)
```

OncoBayes2 *OncoBayes2*

Description

Bayesian logistic regression model with optional EXchangeability-NonEXchangeability parameter modelling for flexible borrowing from historical or concurrent data-sources. The safety model can guide dose-escalation decisions for adaptive Oncology phase I dose-escalation trials which involve an arbitrary number of drugs.

Global Options

Option	Default	Description
`OncoBayes2.MC.warmup`	1000	MCMC warmup iterations
`OncoBayes2.MC.iter`	2000	total MCMC iterations
`OncoBayes2.MC.save_warmup`	TRUE	save warmup samples
`OncoBayes2.MC.chains`	4	MCMC chains
`OncoBayes2.MC.thin`	1	MCMC thinning
`OncoBayes2.MC.control`	`list(adapt_delta=0.99, stepsize=0.1)`	sets `control` argument for Stan call
`OncoBayes2.MC.backend`	rstan	Backend used to run Stan (`rstan` or `cmdstanr`)
`OncoBayes2.abbreviate.min`	0	Minimal length of variable names when abbreviating variable names. The default 0 disables abbreviation.

Author(s)

Maintainer: Sebastian Weber <sebastian.weber@novartis.com>

Authors:

- Sebastian Weber <sebastian.weber@novartis.com>
- Lukas A. Widmer <lukas_andreas.widmer@novartis.com>
- Andrew Bean <andrew.bean@novartis.com>

Other contributors:

- Novartis Pharma AG [copyright holder]
- Trustees of Columbia University (R/stanmodels.R, configure, configure.win) [copyright holder]

References

Neuenschwander B, Roychoudhury S, Schmidli H (2016). "On the use of co-data in clinical trials." *Statistics in Biopharmaceutical Research*, **8**(3), 345–354. doi:10.1080/19466315.2016.1174149.

Neuenschwander B, Wandel S, Roychoudhury S, Bailey S (2016). "Robust exchangeability designs for early phase clinical trials with multiple strata." *Pharmaceutical Statistics*, **15**(2), 123–134. doi:10.1002/pst.1730.

Neuenschwander B, Branson M, Gsponer T (2008). "Critical aspects of the Bayesian approach to phase I cancer trials." *Statistics in Medicine*, **27**(13), 2420–2439. doi:10.1002/sim.3230.

Neuenschwander B, Matano A, Tang Z, Roychoudhury S, Wandel S, Bailey S (2014). "A Bayesian Industry Approach to Phase I Combination Trials in Oncology." In *Statistical Methods in Drug Combination Studies*, volume 69. CRC Press. doi:10.1201/b179659.

Stan Development Team (2024). RStan: the R interface to Stan. R package version 2.32.6. https://mc-stan.org

See Also

Useful links:

- https://opensource.nibr.com/OncoBayes2/
- Report bugs at https://github.com/Novartis/OncoBayes2/issues

---

`plot_blrm` *Plot a fitted model*

Description

Warning: these methods are at an experimental stage of development, and may change with future releases.

Plotting methods for `blrmfit` and `blrm_trial` objects.

Usage

```
plot_toxicity_curve(object, ...)

plot_toxicity_intervals(object, ...)

plot_toxicity_intervals_stacked(object, ...)

## S3 method for class 'blrmfit'
plot_toxicity_curve(
  object,
  newdata,
  x,
  group,
  xlim,
  ylim,
  transform = TRUE,
  prob = 0.5,
  prob_outer = 0.95,
  size = 0.75,
  alpha = 1,
  facet_args = list(),
  hline_at = c(0.16, 0.33),
  grid_length = 100,
  ...
)

## S3 method for class 'blrm_trial'
plot_toxicity_curve(
  object,
  newdata,
```

```
  x,
  group,
  xlim,
  ylim,
  transform = TRUE,
  prob = 0.5,
  prob_outer = 0.95,
  size = 0.75,
  alpha = 1,
  facet_args = list(),
  hline_at,
  grid_length = 100,
  ewoc_shading = TRUE,
  ...
)

## S3 method for class 'blrmfit'
plot_toxicity_intervals(
  object,
  newdata,
  x,
  group,
  interval_prob = c(0, 0.16, 0.33, 1),
  interval_max_mass = c(NA, NA, 0.25),
  ewoc_colors = c("green", "red"),
  facet_args = list(),
  ...
)

## S3 method for class 'blrm_trial'
plot_toxicity_intervals(
  object,
  newdata,
  x,
  group,
  interval_prob,
  interval_max_mass,
  ewoc_colors = c("green", "red"),
  ...
)

## S3 method for class 'blrmfit'
plot_toxicity_intervals_stacked(
  object,
  newdata,
  x,
  group,
  xlim,
```

```
  ylim = c(0, 0.5),
  predictive = FALSE,
  transform = !predictive,
  interval_prob,
  grid_length = 100,
  facet_args = list(),
  ...
)

## S3 method for class 'blrm_trial'
plot_toxicity_intervals_stacked(
  object,
  newdata,
  x,
  group,
  xlim,
  ylim = c(0, 0.5),
  predictive = FALSE,
  transform = !predictive,
  interval_prob,
  grid_length = 100,
  ewoc_shading = TRUE,
  facet_args = list(),
  ...
)
```

Arguments

object	fitted model object
...	currently unused
newdata	optional data frame specifying for what to predict; if missing, then the data of the input model object is used. If object is a blrmfit object, newdata defaults to the data argument. If object is a blrm_trial, it defaults to summary(object, "dose_info").
x	Character giving the parameter name to be mapped to the x-axis. This also supports 'tidy' parameter selection by specifying x = vars(...), where ... is specified the same way as in dplyr::select() and similar functions. Examples of using x in this way can be found in the examples. For blrm_trial methods, it defaults to the first entry in summary(blrm_trial,"drug_info")$drug_name.
group	Grouping variable(s) whose levels will be mapped to different facets of the plot. group can be a character vector, tidy parameter(s) of the form group = vars(...), or a formula to be passed directly to ggplot2::facet_wrap(). For blrm_trial methods, it defaults to group_id, plus all entries of summary(blrm_trial,"drug_info")$d except the first, which is mapped to x.
xlim	x-axis limits
ylim	y-axis limits on the probability scale

transform	logical (defaults to FALSE) indicating if the linear predictor on the logit link scale is transformed with inv_logit to the 0-1 response scale.
prob	central probability mass to report for the inner ribbon, i.e. the quantiles 0.5-prob/2 and 0.5+prob/2 are displayed.
prob_outer	central probability mass to report for the outer ribbon, i.e. the quantiles 0.5-prob/2 and 0.5+prob/2 are displayed.
alpha, size	Arguments passed to geoms. For this plot, alpha is passed to ggplot2::geom_ribbon(), and size is passed to ggplot2::geom_line().
facet_args	A named list of arguments (other than facets) passed to ggplot2::facet_wrap().
hline_at	Location(s) of horizontal guide lines (passed to bayesplot::hline_at()).
grid_length	Number of grid points within xlim for plotting.
ewoc_shading	logical indicates if doses violating EWOC should be shaded in gray. Applies only to blrm_trial methods. Defaults to TRUE.
interval_prob	defines the interval probabilities reported in the standard outputs. Defaults to c(0, 0.16, 0.33, 1), when predictive = FALSE and/or transform = TRUE, or to intervals giving 0, 1, or 2+ DLTs when predictive = TRUE and transform = FALSE. For blrm_trial methods, this is taken from summary(blrm_trial, "interval_prob") by default.
interval_max_mass	vector defining for each interval of the interval_prob vector a maximal admissible probability mass for a given dose level. Whenever the posterior probability mass in a given interval exceeds the threshold, then the Escalation With Overdose Control (EWOC) criterion is considered to be not fulfilled. Dose levels not fulfilling EWOC are ineligible for the next cohort of patients. The default restricts the overdose probability to less than 0.25. For blrm_trial methods, this is taken from summary(blrm_trial, "interval_max_mass") by default.
ewoc_colors	Fill colors used for bars indicating EWOC OK or not. Vector of two characters, each of which must correspond to bayesplot::bayesplot-package() color schemes (see ?[bayesplot::color_scheme_get()][bayesplot::color_scheme_get])
predictive	logical indicates if the posterior predictive is being summarized. Defaults to FALSE.

Details

plot_toxicity_curve plots continuous profiles of the dose-toxicity curve.

plot_toxicity_intervals plots the posterior probability mass in subintervals of $[0, 1]$, at a discrete set of provisional doses.

plot_toxicity_intervals_stacked is similar to plot_toxicity_intervals, but over a continuous range of doses.

Value

A ggplot object that can be further customized using the ggplot2::ggplot2() package.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)


example_model("combo2", silent = TRUE)

# Plot the dose-toxicity curve
plot_toxicity_curve(blrmfit,
  x = "drug_A",
  group = ~ group_id * drug_B,
  newdata = subset(dose_info_combo2, group_id == "trial_AB"),
  facet_args = list(ncol = 4)
)

# Plot posterior DLT-rate-interval probabilities at discrete dose levels
plot_toxicity_intervals(blrmfit,
  x = "drug_A",
  group = ~ group_id * drug_B,
  newdata = subset(dose_info_combo2, group_id == "trial_AB")
)

# Plot posterior DLT-rate-interval probabilities over continuous dose
plot_toxicity_intervals_stacked(blrmfit,
  x = "drug_A",
  group = ~ group_id * drug_B,
  newdata = subset(dose_info_combo2, group_id == "trial_AB")
)

# Plot predictive distribution probabilities over continuous dose
plot_toxicity_intervals_stacked(blrmfit,
  x = "drug_A",
  group = ~ group_id * drug_B,
  predictive = TRUE,
  interval_prob = c(-1, 0, 1, 6),
  newdata = transform(
    subset(
      dose_info_combo2,
      group_id == "trial_AB"
    ),
    num_patients = 6,
    num_toxicities = 0
  )
)
## Recover user set sampling defaults
options(.user_mc_options)
```

`posterior_interval.blrmfit` *Posterior intervals*

Description

Posterior intervals of all model parameters.

Usage

```
## S3 method for class 'blrmfit'
posterior_interval(object, prob = 0.95, ...)
```

Arguments

`object`	fitted model object
`prob`	central probability mass to report, i.e. the quantiles 0.5-prob/2 and 0.5+prob/2 are displayed. Multiple central widths can be specified.
`...`	not used in this function

Details

Reports the quantiles of posterior parameters which correspond to the central probability mass specified. The output includes the posterior of the hyper-parameters and the posterior of each group estimate.

Value

Matrix of two columns for the central probability interval `prob` for all parameters of the model.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

example_model("single_agent", silent = TRUE)

posterior_interval(blrmfit)

## Recover user set sampling defaults
options(.user_mc_options)
```

posterior_linpred.blrmfit *Posterior of linear predictor*

Description

Calculates the posterior of the linear predictor.

Usage

```
## S3 method for class 'blrmfit'
posterior_linpred(object, transform = FALSE, newdata, draws, ...)
```

Arguments

object	fitted model object
transform	logical (defaults to FALSE) indicating if the linear predictor on the logit link scale is transformed with inv_logit to the 0-1 response scale.
newdata	optional data frame specifying for what to predict; if missing, then the data of the input model object is used
draws	number of returned posterior draws; by default the entire posterior is returned
...	not used in this function

Details

Simulates the posterior of the linear predictor of the model object for the specified data set.

Value

Matrix of dimensions draws by nrow(newdata) where row correspond to a draw of the posterior and each column corresponds to a row in newdata. The columns are labelled with the row.names of newdata.

Group and strata definitions

The groups and strata as defined when running the blrm_exnex analysis cannot be changed at a later stage. As a result no evaluations can be performed for groups which have not been present in the data set used for running the analysis. However, it is admissible to code the group (and/or stratum) column as a factor which contains empty levels. These groups are thus not contained in the fitting data set and they are assigned by default to the first stratum. In addition priors must be setup for these groups (and/or strata). These empty group (and/or strata) levels are then allowed in subsequent evaluations. This enables the evaluation of the hierarchical model in terms of representing a prior for future groups.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)


## run single-agent analysis which defines blrmfit model object
example_model("single_agent", silent = TRUE)

## obtain posterior of linear prediction on 0-1 scale
post_prob_dlt <- posterior_linpred(blrmfit, TRUE, newdata = hist_SA)
## name columns to obtain nice bayesplot labels
colnames(post_prob_dlt) <- hist_SA$drug_A

library(bayesplot)
library(ggplot2)
mcmc_intervals(post_prob_dlt, prob = 0.5, prob_outer = 0.95) +
  coord_flip() +
  vline_at(c(0.16, 0.33), linetype = 2) +
  ylab("Dose [mg]") +
  ggtitle("Posterior Probability of a DLT") +
  scale_x_continuous(breaks = c(0.1, 0.16, 0.33, 0.5, 0.75))

## Recover user set sampling defaults
options(.user_mc_options)
```

---

posterior_predict.blrmfit

Posterior of predictive

Description

Simulation of the predictive distribution.

Usage

```
## S3 method for class 'blrmfit'
posterior_predict(object, newdata, draws, ...)
```

Arguments

object	fitted model object
newdata	optional data frame specifying for what to predict; if missing, then the data of the input model object is used
draws	number of returned posterior draws; by default the entire posterior is returned
...	not used in this function

Details

Simulates the posterior predictive of the model `object` for the specified data set.

Value

Matrix of dimensions `draws` by `nrow(newdata)` where row correspond to a draw of the posterior and each column corresponds to a row in `newdata`. The columns are labelled with the `row.names` of `newdata`.

Group and strata definitions

The groups and strata as defined when running the `blrm_exnex` analysis cannot be changed at a later stage. As a result no evaluations can be performed for groups which have not been present in the data set used for running the analysis. However, it is admissible to code the group (and/or stratum) column as a `factor` which contains empty levels. These groups are thus not contained in the fitting data set and they are assigned by default to the first stratum. In addition priors must be setup for these groups (and/or strata). These empty group (and/or strata) levels are then allowed in subsequent evaluations. This enables the evaluation of the hierarchical model in terms of representing a prior for future groups.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)


example_model("single_agent", silent = TRUE)

post_pred <- posterior_predict(blrmfit)
## turn DLT counts into DLT rates
post_pred_rate <- sweep(post_pred, 2, hist_SA$num_patients, "/")

library(bayesplot)
library(ggplot2)

## compare posterior predictive of the model for the response rates
## with observed data
with(
  hist_SA,
 ppc_intervals(num_toxicities / num_patients, post_pred_rate, x = drug_A, prob_outer = 0.95)
) +
  xlab("Dose [mg]")

## Recover user set sampling defaults
options(.user_mc_options)
```

`predictive_interval.blrmfit` *Posterior predictive intervals*

Description

Posterior predictive intervals of the model.

Usage

```
## S3 method for class 'blrmfit'
predictive_interval(object, prob = 0.95, newdata, ...)
```

Arguments

`object`	fitted model object
`prob`	central probability mass to report, i.e. the quantiles 0.5-prob/2 and 0.5+prob/2 are displayed. Multiple central widths can be specified.
`newdata`	optional data frame specifying for what to predict; if missing, then the data of the input model `object` is used
`...`	not used in this function

Details

Reports for each row of the input data set the predictive interval according to the fitted model.

Value

Matrix with as many rows as the input data set and two columns which contain the lower and upper quantile corresponding to the central probability mass `prob` for the number of responses of the predictive distribution.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

example_model("single_agent", silent = TRUE)

predictive_interval(blrmfit)

## Recover user set sampling defaults
options(.user_mc_options)
```

`prior_summary.blrmfit` *Summarise model prior*

Description

Extracts a summary of the prior in a structured data format.

Usage

```
## S3 method for class 'blrmfit'
prior_summary(object, digits = 2, ...)
```

Arguments

`object`	`blrmfit` (`blrm_trial`) object as returned from `blrm_exnex()` (`blrm_trial()`) analysis
`digits`	number of digits to show
`...`	ignored by the function

Details

The summary of the prior creates a structured representation of the specified prior from a `blrm_exnex()` (`blrm_trial()`) analysis.

Value

Returns an analysis specific list, which has it's own `print` function. The returned list contains arrays which represent the prior in a structured format.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

## run combo2 analysis which defines blrmfit model object
example_model("combo2", silent = TRUE)

prior_summary(blrmfit)

prior_sum <- prior_summary(blrmfit)
names(prior_sum)

## the entries of the prior list are labelled arrays
dimnames(prior_sum$EX_mu_log_beta)
```

```
## Recover user set sampling defaults
options(.user_mc_options)
```

`summary.blrmfit` *Summarise model results*

Description

Provides model summaries for `blrm_exnex()` and `blrm_trial()` analyses.

Usage

```
## S3 method for class 'blrmfit'
summary(
  object,
  newdata,
  transform = !predictive,
  prob = 0.95,
  interval_prob,
  predictive = FALSE,
  ...
)
```

Arguments

`object`	fitted model object
`newdata`	optional data frame specifying for what to predict; if missing, then the data of the input model `object` is used
`transform`	logical (defaults to `FALSE`) indicating if the linear predictor on the logit link scale is transformed with `inv_logit` to the 0-1 response scale.
`prob`	central probability mass to report, i.e. the quantiles 0.5-prob/2 and 0.5+prob/2 are displayed. Multiple central widths can be specified.
`interval_prob`	optional vector of sorted quantiles for which the interval probabilities are calculated
`predictive`	logical indicates if the posterior predictive is being summarized. Defaults to `FALSE`.
`...`	not used in this function

Details

The calculated posterior summaries are returned as a `data.frame` and contain optional interval probabilites for the specified vector of sorted quantiles. These summaries are calculated on the response scale by default and can be obtained on the link scale when setting `transform=FALSE`.

When the results are requested for the predictive distribution with `predictive=TRUE`, then the link scale refers to the total counts while the transformed scale divides the (predictive) counts by the number of trials such that results are on the 0-1 scale.

Value

Returns a data.frame of the key summaries of the posterior mean, standard deviation, central probability interval, median and optional interval probabilities. Each row of the data.frame corresponds to the respective input data which is by default the same data set as used for the blrm_exnex() analysis or the data specified in the newdata argument.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

example_model("single_agent", silent = TRUE)

## obtain underdosing (0-0.16], target dosing (0.16-0.33] and
## overdosing (0.33-1] probabilities
summary(blrmfit, interval_prob = c(0, 0.16, 0.33, 1))

## obtain predictive distribution for respective cohorts and
## calculate probability for no event, 1 event or >1 event
## note that this does the calculation for the cohort sizes
## as put into the data-set
summary(blrmfit, interval_prob = c(-1, 0, 1, 10), predictive = TRUE)

## to obtain the predictive for a cohort-size of 6 for all patients
## in the data-set one would need to use the newdata argument, e.g.
summary(blrmfit,
  newdata = transform(hist_SA, num_patients = 6),
  interval_prob = c(-1, 0, 1, 10), predictive = TRUE
)

## Recover user set sampling defaults
options(.user_mc_options)
```

---

summary.blrm_trial *Summarise trial*

Description

Provides model summaries for blrm_trial() analyses.

Usage

```
## S3 method for class 'blrm_trial'
summary(
```

```
  object,
  summarize = c("blrmfit", "blrm_exnex_call", "data", "drug_info", "dose_info",
   "dose_prediction", "data_prediction", "newdata_prediction", "dimensionality",
    "interval_prob", "interval_max_mass", "ewoc_check"),
  ...
)
```

Arguments

`object`	`blrm_trial()` object

`summarize`	one of the following options:
	`blrmfit` summary of the underlying blrmfit object with further arguments ...
	`blrm_exnex_call` blrm_exnex call used to create the `blrmfit` object
	`drug_info` drug_info for the trial, contains drugs, reference doses and units
	`dose_info` dose_info that were defined
	`dose_prediction` prediction for the defined `dose_info`
	`data` data that were observed
	`data_prediction` prediction for the observed data
	`newdata_prediction` prediction for data provided with the `newdata` argument
	`dimensionality` numeric vector with entries `"num_components"`, `"num_interaction_terms"`, `"num_groups"`, `"num_strata"`
	`interval_prob` interval probabilities reported in the standard outputs
	`interval_max_mass` named vector defining for each interval of the `interval_prob` vector a maxmimal admissable probability mass for a given dose level
	`ewoc_check` MCMC diagnostic and precision estimates of ewoc defining metrics for the defined doses in `dose_info` (default) or for the doses provided in the `newdata` argument. Please refer to the details for reported diagnostics.

`...`	further arguments for `summary.blrmfit()`

Details

The `ewoc_check` summary routine allows to assess the accuracy and reliability of the ewoc criterion with respect to MCMC sampling noise. The returned summary provides detailled MCMC convergence and precision estimates for all criteria defined by `interval_prob` and `interval_max_mass` which contribute to EWOC metric. That is, for each interval probability with a maximal mass of less than unity the routine will return these columns:

`est` the MCMC estimate defining the critical value. For intervals defined by a tail probability this corresponds to the respective critical quantile while for interval probabilites this is equal to the interval probability.

`stat` centered and standardized test quantity. The estimate is centered by the critical value and scaled by the Monte-Carlo standard error (MCSE) of the estimate. Hence, negative (positive) values correspond to the constraint being (not) fulfilled. The standardization with the MCSE allows to compare the values to standard normal quantiles accordingly.

`mcse` the Monte-Carlo standard error of the estimate determined with `posterior::mcse_quantile()` (tail probability) or `posterior::mcse_mean()` (interval probability) functions.

`ess` the Monte-Carlo effective sample size of the estimate determined with `posterior::ess_quantile()` (tail probability) or `posterior::ess_mean()` (interval probability) functions.

`rhat` the Monte-Carlo non-convergence diagnostic Rhat as determined with the `[rhat][posterior::rhat] function`.

For the common case of requiring that 33% DLT probability is not exceeded by more than 25% of the posterior probability mass, the estimate column `est` contains the 75% quantile $q_{75\%}$ and the standardized statistic `stat` is defined as:

$$\text{stat} = \frac{q_{75\%} - 33\%}{\text{mcse}_{q_{75\%}}}$$

The statistic is approximately distributed as a standard normal variate. The `ewoc_check` summary can be used to ensure that the MCMC estimation accuracy is sufficient.

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

# construct initial blrm_trial object from built-in example datasets
combo2_trial_setup <- blrm_trial(
  data = hist_combo2,
  dose_info = dose_info_combo2,
  drug_info = drug_info_combo2,
  simplified_prior = TRUE
)

# extract blrm_call to see setup of the prior as passed to blrm_exnex
summary(combo2_trial_setup, "blrm_exnex_call")

# extract ewoc precision accuracy
ec <- summary(combo2_trial_setup, "ewoc_check")

# find any ewoc metrics which are within 95% MCMC error of the threshold
# these are counted as "imprecise" when printing blrm_trial objects
subset(ec, abs(prob_overdose_stat) < qnorm(0.975))

# ensure that the ewoc metric only flags "ok" whenever the MCMC error
# is with 95% below the threshold
ewoc_ok <- ec$prob_overdose_stat < qnorm(0.025)

## Recover user set sampling defaults
options(.user_mc_options)
```

update.blrmfit *Update data of a BLRM analysis*

Description

Adds data rows to a blrm_exnex() or blrm_trial() analysis object.

Usage

```
## S3 method for class 'blrmfit'
update(object, ..., add_data)
```

Arguments

object	blrmfit analysis object
...	passed to default update command The data in add_data will be combined with data in object using bind_rows. The indices for groups and stratums (if defined) are matched between add_data and the data of the analysis object. Note that the add_data argument must be named explicitly as demonstrated in the example.
add_data	additional data added to analysis data of object

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)

example_model("single_agent", silent = TRUE)

library(tibble)
new_cohort <- tibble(group_id = "trial_A", drug_A = 50, num_patients = 4, num_toxicities = 1)

## this would fail, since add_data argument must be named
## new_blrmfit <- update(blrmfit, new_cohort)
new_blrmfit <- update(blrmfit, add_data = new_cohort)

## Recover user set sampling defaults
options(.user_mc_options)
```

`update.blrm_trial` *Update data and/or prior of a BLRM trial*

Description

- Adds data rows to a `blrm_trial()` object (add_data argument)
- Replaces data of a `blrm_trial()` object (data argument)
- Sets the prior of a `blrm_trial()` object (... argument will be passed to blrm_exnex)

Usage

```
## S3 method for class 'blrm_trial'
update(object, ...)
```

Arguments

`object`	blrm_trial object
`...`	passed to default `update` command of `blrm_exnex`

Examples

```
## Setting up dummy sampling for fast execution of example
## Please use 4 chains and 100x more warmup & iter in practice
.user_mc_options <- options(
  OncoBayes2.MC.warmup = 10, OncoBayes2.MC.iter = 20, OncoBayes2.MC.chains = 1,
  OncoBayes2.MC.save_warmup = FALSE
)


# the combo2_trial example demonstrates the use of add_data of
# update.blrmfit
example_model("combo2_trial")

## Recover user set sampling defaults
options(.user_mc_options)
```

Index

∗ **blrm_trial combo2 example**
  blrm_trial, 14
  dose_info_combo2, 24
  drug_info_combo2, 26
  example-combo2_trial, 29
∗ **datasets**
  codata_combo2, 18
  dose_info_combo2, 24
  drug_info_combo2, 26
  hist_combo2, 36
  hist_combo3, 37
  hist_SA, 38

as_draws *(*draws-OncoBayes2*)*, 24
as_draws_array *(*draws-OncoBayes2*)*, 24
as_draws_df *(*draws-OncoBayes2*)*, 24
as_draws_list *(*draws-OncoBayes2*)*, 24
as_draws_matrix *(*draws-OncoBayes2*)*, 24
as_draws_rvars *(*draws-OncoBayes2*)*, 24

bayesplot::hline_at(), *46*
bind_rows_0, 3
blrm_exnex, 4
blrm_exnex(), *29*, *53–55*, *58*
blrm_formula_linear, 12
blrm_formula_saturating, 13
blrm_trial, 14
blrm_trial(), *24*, *26*, *29*, *53–56*, *58*, *59*

codata_combo2, 18
critical_quantile, 19

diagnostic-quantities, 22
dose_info_combo2, *17*, 24, *26*, *29*
dplyr::select(), *20*, *45*
draws-OncoBayes2, 24
drug_info_combo2, *17*, *24*, 26, *29*

example-combo2, 27
example-combo2_trial, 29
example-combo3, 31
example-single-agent, 34
example_model, 35
example_model(), *29*

ggplot2::facet_wrap(), *45*, *46*
ggplot2::geom_line(), *46*
ggplot2::geom_ribbon(), *46*
ggplot2::ggplot2(), *46*

hist_combo2, 36
hist_combo3, 37
hist_SA, 38

inv_logit *(*lodds*)*, 40

lodds, 40
log_posterior *(*diagnostic-quantities*)*, 22
logit *(*lodds*)*, 40

neff_ratio *(*diagnostic-quantities*)*, 22
nsamples *(*nsamples.blrmfit*)*, 41
nsamples.blrmfit, 41
nuts_params *(*diagnostic-quantities*)*, 22

OncoBayes2, 41
OncoBayes2-package *(*OncoBayes2*)*, 41

plot_blrm, 43
plot_toxicity_curve *(*plot_blrm*)*, 43
plot_toxicity_intervals *(*plot_blrm*)*, 43
plot_toxicity_intervals_stacked *(*plot_blrm*)*, 43
posterior::draws(), *25*
posterior::ess_mean(), *57*
posterior::ess_quantile(), *57*
posterior::mcse_mean(), *56*
posterior::mcse_quantile(), *56*
posterior::subset_draws(), *25*
posterior_interval *(*posterior_interval.blrmfit*)*, 48

posterior_interval.blrmfit, 48
posterior_linpred
(posterior_linpred.blrmfit), 49
posterior_linpred.blrmfit, 49
posterior_predict
(posterior_predict.blrmfit), 50
posterior_predict.blrmfit, 50
predictive_interval
(predictive_interval.blrmfit),
52
predictive_interval.blrmfit, 52
print.blrm_trial (blrm_trial), 14
print.blrmfit (blrm_exnex), 4
prior_summary (prior_summary.blrmfit),
53
prior_summary.blrmfit, 53

rhat (diagnostic-quantities), 22

stan, *7*
stats::uniroot(), *21*
summary.blrm_trial, 55
summary.blrm_trial(), *29*
summary.blrmfit, 54
summary.blrmfit(), *56*

update.blrm_trial, 59
update.blrm_trial(), *29*
update.blrmfit, 58