Package 'beaver'

May 22, 2024

Title Bayesian Model Averaging of Covariate Adjusted Negative-Binomial Dose-Response

Version 1.0.0

Description Dose-response modeling for negative-binomial distributed data with a variety of dose-response models. Covariate adjustment and Bayesian model averaging is supported. Functions are provided to easily obtain inference on the dose-response relationship and plot the dose-response curve.

License MIT + file LICENSE

URL https://github.com/rich-payne/beaver

Depends R (>= 3.5.0)

Imports checkmate (>= 2.1), dplyr (>= 1.0), ellipsis (>= 0.3), fs (>= 1.5), ggplot2 (>= 3.3), purrr (>= 0.3), rjags (>= 4.12), rlang (>= 1.0), stringr (>= 1.5), tibble (>= 3.1), tidyr (>= 1.1), yodel (>= 1.0)

Encoding UTF-8

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Richard Payne [aut],
Hollins Showalter [aut, cre],
Eli Lilly and Company [cph]

Maintainer Hollins Showalter <`hollins.showalter@gmail.com`>

Repository CRAN

Date/Publication 2024-05-22 13:00:06 UTC

R topics documented:

`beaver_mcmc` *Bayesian Model Averaging of Covariate Adjusted Neg-Binomial Dose-Response*

Description

Bayesian Model Averaging of Covariate Adjusted Neg-Binomial Dose-Response

Usage

```
beaver_mcmc(
  data,
  formula = ~1,
  ...,
  n_adapt = 1000,
  n_burn = 1000,
  n_iter = 10000,
  n_chains = 4,
  thin = 1,
  quiet = FALSE
)
```

Arguments

`data`	a dataframe with columns "dose", "response" and any covariates listed in the `formula` argument.
`formula`	a right-hand sided formula specifying the covariates.
`...`	candidate models to be included in Bayesian model averaging. These should be created from calls to the `model_negbin_*` functions (e.g. `model_negbin_emax()`).
`n_adapt`	the number of iterations used to tune the MCMC algorithm.

`n_burn`	the number of MCMC iterations used for burn-in.
`n_iter`	the number of MCMC iterations to save.
`n_chains`	the number of MCMC chains.
`thin`	thinning for the MCMC chain.
`quiet`	logical indicating if MCMC chain progress output should be silenced.

Value

A list (with appropriate S3 classes) with the prior and posterior weights, sampled model index, and individual MCMC fits.

See Also

Other models: `model_negbin_emax()`, `model_negbin_exp()`, `model_negbin_indep()`, `model_negbin_linear()`, `model_negbin_loglinear()`, `model_negbin_logquad()`, `model_negbin_quad()`, `model_negbin_sigmoid_emax()`

Other posterior calculations: `posterior.beaver_mcmc_bma()`, `posterior.beaver_mcmc()`, `posterior_g_comp()`, `pr_eoi_g_comp()`, `pr_eoi()`

Examples

```
# The {beaver} package, by definition, performs MCMC for multiple models.
# Even with a small number of chains/burn-ins/samples, a minimally illustrative
# example requires >5s to run.

library(dplyr)

# No covariates----

set.seed(100)

df <- data_negbin_emax(
  n_per_arm = 10,
  doses = 0:3,
  b1 = 0,
  b2 = 2.5,
  b3 = 0.5,
  ps = 0.75
)

df %>%
  group_by(dose) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc <- beaver_mcmc(
  emax = model_negbin_emax(
```

```
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  linear = model_negbin_linear(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    w_prior = 1 / 4
  ),
  quad = model_negbin_quad(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ 1,
  data = df,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc$w_post

draws <- try(draws(mcmc)) #draws() is intended for single model fits only
draws_emax <- draws(mcmc$models$emax$mcmc)
draws_linear <- draws(mcmc$models$linear$mcmc)
draws_quad <- draws(mcmc$models$quad$mcmc)
draws_exp <- draws(mcmc$models$exp$mcmc)

post <- posterior(
  mcmc,
  contrast = matrix(1, 1, 1),
  doses = 0:3,
  reference_dose = 0,
```

```
  reference_type = "difference"
)

pr_eoi(
  mcmc,
  eoi = c(5, 8),
  contrast = matrix(1, 1, 1),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp <- posterior_g_comp(
  mcmc,
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc,
  eoi = c(5, 8),
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc, contrast = matrix(1, 1, 1))

# With covariates----

set.seed(1000)

x <-
  data.frame(
    gender = factor(sample(c("F", "M"), 40, replace = TRUE))
  ) %>%
  model.matrix(~ gender, data = .)

df_cov <-
  data_negbin_emax(
    n_per_arm = 10,
    doses = 0:3,
    b1 = c(0, 0.5),
    b2 = 2.5,
    b3 = 0.5,
    ps = 0.75,
    x = x
  ) %>%
  mutate(
    gender = case_when(
      genderM == 1 ~ "M",
      TRUE ~ "F"
    ),
```

```
    gender = factor(gender)
  ) %>%
  select(subject, dose, gender, response)

df_cov %>%
  group_by(dose, gender) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc_cov <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  linear = model_negbin_linear(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    w_prior = 1 / 4
  ),
  quad = model_negbin_quad(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ gender,
  data = df_cov,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)
```

```
mcmc_cov$w_post

draws_cov <- try(draws(mcmc_cov)) #draws() is intended for single model fits only
draws_cov_emax <- draws(mcmc_cov$models$emax$mcmc)
draws_cov_linear <- draws(mcmc_cov$models$linear$mcmc)
draws_cov_quad <- draws(mcmc_cov$models$quad$mcmc)
draws_cov_exp <- draws(mcmc_cov$models$exp$mcmc)

post_cov <- posterior(
  mcmc_cov,
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc_cov,
  eoi = c(5, 8),
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp_cov <- posterior_g_comp(
  mcmc_cov,
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc_cov,
  eoi = c(5, 8),
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc_cov, new_data = df_cov, type = "g-comp")
```

`data_negbin_emax` *Generate data from a negative binomial EMAX model*

Description

Generate data from a negative binomial EMAX model

Usage

```
data_negbin_emax(n_per_arm, doses, b1, b2, b3, ps, x = NULL)
```

Arguments

`n_per_arm`	number of subjects in each dose arm.
`doses`	doses at which to simulate subjects.
`b1, b2, b3, ps`	parameters from which to simulate data. See model description below. If covariates are specified (through `x`), then `b1` should be a vector of length `ncol(x)`.
`x`	the model matrix for the covariates. Must have the same number of rows as the total number of subjects (`sum(n_per_arm * rep(1, length(doses)))`). If `NULL`, then an intercept term is used by default.

Value

A dataframe with columns "subject", "dose", and "response".

Negative Binomial EMAX

Let y_{ij} be the jth subject on dose d_i. The model is

$$y_{ij}\ NB(p_i, r_i)$$

$$p_i\ Uniform(0, 1)$$

$$r_{ij} = (\mu_{ij} * p_i)/(1 - p_i)$$

$$log(\mu_{ij}) = x_{ij} * b1 + b2 * d_i/(b3 + d_i)$$

$$b1\ N(`mu_b1`, `sigma_b1`^2)$$

$$b2\ N(`mu_b2`, `sigma_b2`^2)$$

$$b3\ N(`mu_b3`, `sigma_b3`^2)(Truncatedtobepositive)$$

The model is parameterized in terms of the mean of the negative binomial distribution and the usual probability parameter p. The prior on the mean is an EMAX model, and the prior on p at each dose is Uniform(0, 1). The model can adjust for baseline covariates, (

$$x_{ij}$$

).

Examples

```
# The {beaver} package, by definition, performs MCMC for multiple models.
# Even with a small number of chains/burn-ins/samples, a minimally illustrative
# example requires >5s to run.

library(dplyr)

# No covariates----
```

```
set.seed(100)

df <- data_negbin_emax(
  n_per_arm = 10,
  doses = 0:3,
  b1 = 0,
  b2 = 2.5,
  b3 = 0.5,
  ps = 0.75
)

df %>%
  group_by(dose) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  linear = model_negbin_linear(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    w_prior = 1 / 4
  ),
  quad = model_negbin_quad(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
```

```
    w_prior = 1 / 4
  ),
  formula = ~ 1,
  data = df,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc$w_post

draws <- try(draws(mcmc)) #draws() is intended for single model fits only
draws_emax <- draws(mcmc$models$emax$mcmc)
draws_linear <- draws(mcmc$models$linear$mcmc)
draws_quad <- draws(mcmc$models$quad$mcmc)
draws_exp <- draws(mcmc$models$exp$mcmc)

post <- posterior(
  mcmc,
  contrast = matrix(1, 1, 1),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc,
  eoi = c(5, 8),
  contrast = matrix(1, 1, 1),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp <- posterior_g_comp(
  mcmc,
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc,
  eoi = c(5, 8),
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc, contrast = matrix(1, 1, 1))

# With covariates----

set.seed(1000)
```

```
x <-
  data.frame(
    gender = factor(sample(c("F", "M"), 40, replace = TRUE))
  ) %>%
  model.matrix(~ gender, data = .)

df_cov <-
  data_negbin_emax(
    n_per_arm = 10,
    doses = 0:3,
    b1 = c(0, 0.5),
    b2 = 2.5,
    b3 = 0.5,
    ps = 0.75,
    x = x
  ) %>%
  mutate(
    gender = case_when(
      genderM == 1 ~ "M",
      TRUE ~ "F"
    ),
    gender = factor(gender)
  ) %>%
  select(subject, dose, gender, response)

df_cov %>%
  group_by(dose, gender) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc_cov <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  linear = model_negbin_linear(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    w_prior = 1 / 4
  ),
  quad = model_negbin_quad(
    mu_b1 = 0,
```

```
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ gender,
  data = df_cov,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc_cov$w_post

draws_cov <- try(draws(mcmc_cov)) #draws() is intended for single model fits only
draws_cov_emax <- draws(mcmc_cov$models$emax$mcmc)
draws_cov_linear <- draws(mcmc_cov$models$linear$mcmc)
draws_cov_quad <- draws(mcmc_cov$models$quad$mcmc)
draws_cov_exp <- draws(mcmc_cov$models$exp$mcmc)

post_cov <- posterior(
  mcmc_cov,
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc_cov,
  eoi = c(5, 8),
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp_cov <- posterior_g_comp(
  mcmc_cov,
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)
```

```
pr_eoi_g_comp(
  mcmc_cov,
  eoi = c(5, 8),
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc_cov, new_data = df_cov, type = "g-comp")
```

`draws` *Posterior Draws*

Description

Extracts posterior draws and puts them into a dataframe or tibble.

Usage

```
draws(x, ...)

## S3 method for class 'beaver_mcmc'
draws(x, ...)

## S3 method for class 'beaver_mcmc_bma'
draws(x, ...)
```

Arguments

`x`	MCMC output.
`...`	additional arguments passed to methods.

Value

For generic: See specific method.

For class 'beaver_mcmc': A dataframe or tibble of MCMC draws.

For class 'beaver_mcmc_bma': An error.

Examples

```
# The {beaver} package, by definition, performs MCMC for multiple models.
# Even with a small number of chains/burn-ins/samples, a minimally illustrative
# example requires >5s to run.
```

```
library(dplyr)

# No covariates----

set.seed(100)

df <- data_negbin_emax(
  n_per_arm = 10,
  doses = 0:3,
  b1 = 0,
  b2 = 2.5,
  b3 = 0.5,
  ps = 0.75
)

df %>%
  group_by(dose) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  linear = model_negbin_linear(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    w_prior = 1 / 4
  ),
  quad = model_negbin_quad(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
```

```
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ 1,
  data = df,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc$w_post

draws <- try(draws(mcmc)) #draws() is intended for single model fits only
draws_emax <- draws(mcmc$models$emax$mcmc)
draws_linear <- draws(mcmc$models$linear$mcmc)
draws_quad <- draws(mcmc$models$quad$mcmc)
draws_exp <- draws(mcmc$models$exp$mcmc)

post <- posterior(
  mcmc,
  contrast = matrix(1, 1, 1),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc,
  eoi = c(5, 8),
  contrast = matrix(1, 1, 1),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp <- posterior_g_comp(
  mcmc,
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc,
  eoi = c(5, 8),
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc, contrast = matrix(1, 1, 1))
```

```
# With covariates----

set.seed(1000)

x <-
  data.frame(
    gender = factor(sample(c("F", "M"), 40, replace = TRUE))
  ) %>%
  model.matrix(~ gender, data = .)

df_cov <-
  data_negbin_emax(
    n_per_arm = 10,
    doses = 0:3,
    b1 = c(0, 0.5),
    b2 = 2.5,
    b3 = 0.5,
    ps = 0.75,
    x = x
  ) %>%
  mutate(
    gender = case_when(
      genderM == 1 ~ "M",
      TRUE ~ "F"
    ),
    gender = factor(gender)
  ) %>%
  select(subject, dose, gender, response)

df_cov %>%
  group_by(dose, gender) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc_cov <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  linear = model_negbin_linear(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    w_prior = 1 / 4
```

```
  ),
  quad = model_negbin_quad(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ gender,
  data = df_cov,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc_cov$w_post

draws_cov <- try(draws(mcmc_cov)) #draws() is intended for single model fits only
draws_cov_emax <- draws(mcmc_cov$models$emax$mcmc)
draws_cov_linear <- draws(mcmc_cov$models$linear$mcmc)
draws_cov_quad <- draws(mcmc_cov$models$quad$mcmc)
draws_cov_exp <- draws(mcmc_cov$models$exp$mcmc)

post_cov <- posterior(
  mcmc_cov,
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc_cov,
  eoi = c(5, 8),
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp_cov <- posterior_g_comp(
  mcmc_cov,
  new_data = df_cov,
```

```
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc_cov,
  eoi = c(5, 8),
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc_cov, new_data = df_cov, type = "g-comp")
```

`model_negbin_emax` *Negative Binomial EMAX Dose Response*

Description

Model settings for a negative binomial distribution assuming an EMAX Model on the mean. This function is to be used within a call to `beaver_mcmc()`.

Usage

```
model_negbin_emax(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  w_prior = 1
)
```

Arguments

`mu_b1, sigma_b1, mu_b2, sigma_b2, mu_b3, sigma_b3`	hyperparameters. See the model description below for context.
`w_prior`	the prior weight for the model.

Value

A list with the model's prior weight and hyperparameter values.

Negative Binomial EMAX

Let y_{ij} be the jth subject on dose d_i. The model is

$$y_{ij}\ NB(p_i, r_i)$$

$$p_i\ Uniform(0, 1)$$

$$r_{ij} = (\mu_{ij} * p_i)/(1 - p_i)$$

$$log(\mu_{ij}) = x_{ij} * b1 + b2 * d_i/(b3 + d_i)$$

$$b1\ N(`mu_b1`, `sigma_b1`^2)$$

$$b2\ N(`mu_b2`, `sigma_b2`^2)$$

$$b3\ N(`mu_b3`, `sigma_b3`^2)(Truncatedtobepositive)$$

The model is parameterized in terms of the mean of the negative binomial distribution and the usual probability parameter p. The prior on the mean is an EMAX model, and the prior on p at each dose is Uniform(0, 1). The model can adjust for baseline covariates, (

$$x_{ij}$$

).

See Also

Other models: `beaver_mcmc()`, `model_negbin_exp()`, `model_negbin_indep()`, `model_negbin_linear()`, `model_negbin_loglinear()`, `model_negbin_logquad()`, `model_negbin_quad()`, `model_negbin_sigmoid_emax()`

`model_negbin_exp` *Negative Binomial Exponential Dose Response*

Description

Model settings for a negative binomial distribution assuming an exponential model on the mean. This function is to be used within a call to `beaver_mcmc()`.

Usage

```
model_negbin_exp(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  w_prior = 1
)
```

Arguments

`mu_b1, sigma_b1, mu_b2, sigma_b2, mu_b3, sigma_b3`	hyperparameters. See the model description below for context.
`w_prior`	the prior weight for the model.

Value

A list with the model's prior weight and hyperparameter values.

Negative Binomial Exponential

Let y_{ij} be the jth subject on dose d_i. The model is

$$y_{ij}\ NB(p_i, r_i)$$

$$p_i\ Uniform(0, 1)$$

$$r_{ij} = (\mu_{ij} * p_i)/(1 - p_i)$$

$$log(\mu_{ij}) = x_{ij} * b1 + b2 * (1 - exp(-b3 * d_i))$$

$$b1\ N(`mu_b1`, `sigma_b1`^2)$$

$$b2\ N(`mu_b2`, `sigma_b2`^2)$$

$$b3\ N(`mu_b3`, `sigma_b3`^2)(Truncatedtobepositive)$$

The model is parameterized in terms of the mean of the negative binomial distribution and the usual probability parameter p. The prior on the mean is an exponential model, and the prior on p at each dose is Uniform(0, 1). The model can adjust for baseline covariates, (

$$x_{ij}$$

).

See Also

Other models: `beaver_mcmc()`, `model_negbin_emax()`, `model_negbin_indep()`, `model_negbin_linear()`, `model_negbin_loglinear()`, `model_negbin_logquad()`, `model_negbin_quad()`, `model_negbin_sigmoid_emax()`

`model_negbin_indep` *Negative Binomial Independent Dose Response*

Description

Model settings for a negative binomial distribution with an independent mean for each dose. This function is to be used within a call to `beaver_mcmc()`.

Usage

```
model_negbin_indep(mu_b1, sigma_b1, mu_b2, sigma_b2, w_prior = 1)
```

Arguments

`mu_b1`, `sigma_b1`, `mu_b2`, `sigma_b2`	hyperparameters. See the model description below for context.
`w_prior`	the prior weight for the model.

Value

A list with the model's prior weight and hyperparameter values.

Negative Binomial Independent

Let y_{ij} be the jth subject on the kth dose. The model is

$$y_{ij} \, NB(p_i, r_i)$$

$$p_i \, Uniform(0, 1)$$

$$r_{ij} = (\mu_{ij} * p_i)/(1 - p_i)$$

$$log(\mu_{ij}) = x_{ij} * b1 + b2_k$$

$$b1 \, N(`mu_b1`, `sigma_b1`^2)$$

$$b2_k \, N(`mu_b2`, `sigma_b2`^2)$$

The model is parameterized in terms of the mean of the negative binomial distribution and the usual probability parameter p. The prior on the mean is an exponential model, and the prior on p at each dose is Uniform(0, 1). The model can adjust for baseline covariates, (

$$x_{ij}$$

).

See Also

Other models: `beaver_mcmc()`, `model_negbin_emax()`, `model_negbin_exp()`, `model_negbin_linear()`, `model_negbin_loglinear()`, `model_negbin_logquad()`, `model_negbin_quad()`, `model_negbin_sigmoid_emax()`

`model_negbin_linear` *Negative Binomial Linear Dose Response*

Description

Model settings for a negative binomial distribution assuming an linear model on the mean. This function is to be used within a call to `beaver_mcmc()`.

Usage

```
model_negbin_linear(mu_b1, sigma_b1, mu_b2, sigma_b2, w_prior = 1)
```

Arguments

`mu_b1, sigma_b1, mu_b2, sigma_b2`	hyperparameters. See the model description below for context.
`w_prior`	the prior weight for the model.

Value

A list with the model's prior weight and hyperparameter values.

Negative Binomial Linear

Let y_{ij} be the jth subject on dose d_i. The model is

$$y_{ij}\ NB(p_i, r_i)$$

$$p_i\ Uniform(0, 1)$$

$$r_{ij} = (\mu_{ij} * p_i)/(1 - p_i)$$

$$log(\mu_{ij}) = x_{ij} * b1 + b2 * d_i$$

$$b1\ N(`mu_b 1`, `sigma_b 1`^2)$$

$$b2\ N(`mu_b 2`, `sigma_b 2`^2)$$

The model is parameterized in terms of the mean of the negative binomial distribution and the usual probability parameter p. The prior on the mean is a linear model, and the prior on p at each dose is Uniform(0, 1). The model can adjust for baseline covariates, (

$$x_{ij}$$

).

See Also

Other models: `beaver_mcmc()`, `model_negbin_emax()`, `model_negbin_exp()`, `model_negbin_indep()`, `model_negbin_loglinear()`, `model_negbin_logquad()`, `model_negbin_quad()`, `model_negbin_sigmoid_emax()`

`model_negbin_loglinear`

Negative Binomial Log-Linear Dose Response

Description

Model settings for a negative binomial distribution assuming a log-linear model on the mean. This function is to be used within a call to `beaver_mcmc()`.

Usage

```
model_negbin_loglinear(mu_b1, sigma_b1, mu_b2, sigma_b2, w_prior = 1)
```

Arguments

`mu_b1, sigma_b1, mu_b2, sigma_b2`	hyperparameters. See the model description below for context.
`w_prior`	the prior weight for the model.

Value

A list with the model's prior weight and hyperparameter values.

Negative Binomial Log-Linear

Let y_{ij} be the jth subject on dose d_i. The model is

$$y_{ij}\ NB(p_i, r_i)$$

$$p_i\ Uniform(0, 1)$$

$$r_{ij} = (\mu_{ij} * p_i)/(1 - p_i)$$

$$log(\mu_{ij}) = x_{ij} * b1 + b2 * log(1 + d_i)$$

$$b1\ N(‘mu_b1‘, ‘sigma_b1‘^2)$$

$$b2\ N(‘mu_b2‘, ‘sigma_b2‘^2)$$

The model is parameterized in terms of the mean of the negative binomial distribution and the usual probability parameter p. The prior on the mean is a log-linear model, and the prior on p at each dose is Uniform(0, 1). The model can adjust for baseline covariates, (

$$x_{ij}$$

).

See Also

Other models: `beaver_mcmc()`, `model_negbin_emax()`, `model_negbin_exp()`, `model_negbin_indep()`, `model_negbin_linear()`, `model_negbin_logquad()`, `model_negbin_quad()`, `model_negbin_sigmoid_emax()`

`model_negbin_logquad` *Negative Binomial Log-Quadratic Dose Response*

Description

Model settings fora negative binomial distribution assuming a log-quadratic model on the mean. This function is to be used within a call to `beaver_mcmc()`.

Usage

```
model_negbin_logquad(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  w_prior = 1
)
```

Arguments

`mu_b1`, `sigma_b1`, `mu_b2`, `sigma_b2`, `mu_b3`, `sigma_b3`	hyperparameters. See the model description below for context.
`w_prior`	the prior weight for the model.

Value

A list with the model's prior weight and hyperparameter values.

Negative Binomial Quadratic

Let y_{ij} be the jth subject on dose d_i. The model is

$$y_{ij}\ NB(p_i, r_i)$$

$$p_i\ Uniform(0, 1)$$

$$r_{ij} = (\mu_{ij} * p_i)/(1 - p_i)$$

$$log(\mu_{ij}) = x_{ij} * b1 + b2 * log(1 + d_i) + b3 * log(1 + d_i)^2$$

$$b1\ N(`mu_b1`, `sigma_b1`^2)$$

$$b2\ N(`mu_b2`, `sigma_b2`^2)$$

$$b3\ N(`mu_b3`, `sigma_b3`^2)$$

The model is parameterized in terms of the mean of the negative binomial distribution and the usual probability parameter p. The prior on the mean is a quadratic model, and the prior on p at each dose is Uniform(0, 1). The model can adjust for baseline covariates, (

$$x_{ij}$$

).

See Also

Other models: `beaver_mcmc()`, `model_negbin_emax()`, `model_negbin_exp()`, `model_negbin_indep()`, `model_negbin_linear()`, `model_negbin_loglinear()`, `model_negbin_quad()`, `model_negbin_sigmoid_emax()`

`model_negbin_quad` *Negative Binomial Quadratic Dose Response*

Description

Model settings for a negative binomial distribution assuming an quadratic model on the mean. This function is to be used within a call to `beaver_mcmc()`.

Usage

```
model_negbin_quad(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  w_prior = 1
)
```

Arguments

`mu_b1`, `sigma_b1`, `mu_b2`, `sigma_b2`, `mu_b3`, `sigma_b3`	hyperparameters. See the model description below for context.
`w_prior`	the prior weight for the model.

Value

A list with the model's prior weight and hyperparameter values.

Negative Binomial Quadratic

Let y_{ij} be the jth subject on dose d_i. The model is

$$y_{ij}\ NB(p_i, r_i)$$

$$p_i\ Uniform(0, 1)$$

$$r_{ij} = (\mu_{ij} * p_i)/(1 - p_i)$$

$$log(\mu_{ij}) = x_{ij} * b1 + b2 * d_i + b3 * d_i^2$$

$$b1\ N(`mu_b1`, `sigma_b1`^2)$$

$$b2\ N(`mu_b2`, `sigma_b2`^2)$$

$$b3\ N(`mu_b3`, `sigma_b3`^2)$$

The model is parameterized in terms of the mean of the negative binomial distribution and the usual probability parameter p. The prior on the mean is a quadratic model, and the prior on p at each dose is Uniform(0, 1). The model can adjust for baseline covariates, (

$$x_{ij}$$

).

See Also

Other models: `beaver_mcmc()`, `model_negbin_emax()`, `model_negbin_exp()`, `model_negbin_indep()`, `model_negbin_linear()`, `model_negbin_loglinear()`, `model_negbin_logquad()`, `model_negbin_sigmoid_emax()`

---

model_negbin_sigmoid_emax

Negative Binomial Sigmoidal EMAX Dose Response

Description

Model settings for a negative binomial distribution assuming a Sigmoidal EMAX Model on the mean. This function is to be used within a call to `beaver_mcmc()`.

Usage

```
model_negbin_sigmoid_emax(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  mu_b4,
  sigma_b4,
  w_prior = 1
)
```

Arguments

`mu_b1, sigma_b1, mu_b2, sigma_b2, mu_b3, sigma_b3, mu_b4, sigma_b4`	hyperparameters. See the model description below for context.
`w_prior`	the prior weight for the model.

Value

A list with the model's prior weight and hyperparameter values.

Negative Binomial Sigmoidal EMAX

Let y_{ij} be the jth subject on dose d_i. The model is

$$y_{ij}\ NB(p_i, r_i)$$

$$p_i\ Uniform(0, 1)$$

$$r_{ij} = (\mu_{ij} * p_i)/(1 - p_i)$$

$$log(\mu_{ij}) = x_{ij} * b1 + b2 * d_i^b 4/(b3 + d_i^b 4)$$

$$b1\ N(`mu_b 1`, `sigma_b 1`^2)$$

$$b2\ N(`mu_b 2`, `sigma_b 2`^2)$$

$$b3\ N(`mu_b 3`, `sigma_b 3`^2)(Truncatedtobepositive)$$

$$b3\ N(`mu_b 4`, `sigma_b 4`^2)(Truncatedtobepositive)$$

The model is parameterized in terms of the mean of the negative binomial distribution and the usual probability parameter p. The prior on the mean is an EMAX model, and the prior on p at each dose is Uniform(0, 1). The model can adjust for baseline covariates, (

$$x_{ij}$$

).

See Also

Other models: `beaver_mcmc()`, `model_negbin_emax()`, `model_negbin_exp()`, `model_negbin_indep()`, `model_negbin_linear()`, `model_negbin_loglinear()`, `model_negbin_logquad()`, `model_negbin_quad()`

`posterior.beaver_mcmc` *Posterior Samples from Bayesian Model Averaging*

Description

Calculate posterior quantities of interest using Bayesian model averaging.

Usage

```
## S3 method for class 'beaver_mcmc'
posterior(
  x,
  doses = attr(x, "doses"),
  reference_dose = NULL,
  prob = c(0.025, 0.975),
  return_stats = TRUE,
  return_samples = FALSE,
  new_data = NULL,
  contrast = NULL,
  reference_type = c("difference", "ratio"),
  ...
)
```

Arguments

`x`	an object output from (internal function) `run_mcmc()`.
`doses`	doses at which to obtain the posterior.
`reference_dose`	dose to which to compare as either a difference or ratio.
`prob`	the percentiles of the posterior to calculate for each dose.
`return_stats`	logical indicating if the posterior mean and quantiles should be returned.
`return_samples`	logical indicating if posterior mean samples should be returned.
`new_data`	a dataframe for which the posterior will be calculated for each observation's covariate values.
`contrast`	a matrix containing where each row contains a contrast for which the posterior will be calculated.
`reference_type`	whether to provide the posterior of the difference or the ratio between each dose and the reference dose.
`...`	additional arguments will throw an error.

Value

A list with the elements `stats` and `samples`. When using this function with default settings, `samples` is NULL and `stats` is a dataframe summarizing the posterior samples. `stats` contains, at a minimum, the columns "dose", ".contrast_index", "(Intercept)", "value", and variables corresponding to the values passed in `prob` ("2.50%" and "97.50%" by default). When `return_stats` is set to `FALSE`, `stats` is NULL. When `return_samples` is set to `TRUE`, `samples` is a dataframe with the posterior samples for each iteration of the MCMC. The dataframe will have, at a minimum, the column "iter", indicating the MCMC iteration, as well as the columns "dose", ".contrast_index", "(Intercept)", and "value". The functions used for each model are defined within the `model_negbin_XYZ()` functions and used in the `run_mcmc()` function.

See Also

Other posterior calculations: `beaver_mcmc()`, `posterior.beaver_mcmc_bma()`, `posterior_g_comp()`, `pr_eoi_g_comp()`, `pr_eoi()`

Examples

```
# The {beaver} package, by definition, performs MCMC for multiple models.
# Even with a small number of chains/burn-ins/samples, a minimally illustrative
# example requires >5s to run.

library(dplyr)

# No covariates----

set.seed(100)

df <- data_negbin_emax(
  n_per_arm = 10,
```

```
    doses = 0:3,
    b1 = 0,
    b2 = 2.5,
    b3 = 0.5,
    ps = 0.75
  )

  df %>%
    group_by(dose) %>%
    summarize(
      mean = mean(response),
      se = sd(response) / sqrt(n()),
      .groups = "drop"
    )

  mcmc <- beaver_mcmc(
    emax = model_negbin_emax(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    linear = model_negbin_linear(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      w_prior = 1 / 4
    ),
    quad = model_negbin_quad(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    exp = model_negbin_exp(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 0,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    formula = ~ 1,
    data = df,
    n_iter = 1e2,
```

```
  n_chains = 1,
  quiet = TRUE
)

mcmc$w_post

draws <- try(draws(mcmc)) #draws() is intended for single model fits only
draws_emax <- draws(mcmc$models$emax$mcmc)
draws_linear <- draws(mcmc$models$linear$mcmc)
draws_quad <- draws(mcmc$models$quad$mcmc)
draws_exp <- draws(mcmc$models$exp$mcmc)

post <- posterior(
  mcmc,
  contrast = matrix(1, 1, 1),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc,
  eoi = c(5, 8),
  contrast = matrix(1, 1, 1),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp <- posterior_g_comp(
  mcmc,
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc,
  eoi = c(5, 8),
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc, contrast = matrix(1, 1, 1))

# With covariates----

set.seed(1000)

x <-
  data.frame(
    gender = factor(sample(c("F", "M"), 40, replace = TRUE))
  ) %>%
```

```
    model.matrix(~ gender, data = .)

  df_cov <-
    data_negbin_emax(
      n_per_arm = 10,
      doses = 0:3,
      b1 = c(0, 0.5),
      b2 = 2.5,
      b3 = 0.5,
      ps = 0.75,
      x = x
    ) %>%
    mutate(
      gender = case_when(
        genderM == 1 ~ "M",
        TRUE ~ "F"
      ),
      gender = factor(gender)
    ) %>%
    select(subject, dose, gender, response)

  df_cov %>%
    group_by(dose, gender) %>%
    summarize(
      mean = mean(response),
      se = sd(response) / sqrt(n()),
      .groups = "drop"
    )

  mcmc_cov <- beaver_mcmc(
    emax = model_negbin_emax(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    linear = model_negbin_linear(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      w_prior = 1 / 4
    ),
    quad = model_negbin_quad(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
```

```
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ gender,
  data = df_cov,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc_cov$w_post

draws_cov <- try(draws(mcmc_cov)) #draws() is intended for single model fits only
draws_cov_emax <- draws(mcmc_cov$models$emax$mcmc)
draws_cov_linear <- draws(mcmc_cov$models$linear$mcmc)
draws_cov_quad <- draws(mcmc_cov$models$quad$mcmc)
draws_cov_exp <- draws(mcmc_cov$models$exp$mcmc)

post_cov <- posterior(
  mcmc_cov,
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc_cov,
  eoi = c(5, 8),
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp_cov <- posterior_g_comp(
  mcmc_cov,
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc_cov,
  eoi = c(5, 8),
  new_data = df_cov,
```

```
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc_cov, new_data = df_cov, type = "g-comp")
```

---

posterior.beaver_mcmc_bma

Posterior Samples from Bayesian Model Averaging

Description

Calculate posterior quantities of interest using Bayesian model averaging.

Usage

```
## S3 method for class 'beaver_mcmc_bma'
posterior(
  x,
  doses = attr(x, "doses"),
  reference_dose = NULL,
  prob = c(0.025, 0.975),
  return_stats = TRUE,
  return_samples = FALSE,
  new_data = NULL,
  contrast = NULL,
  reference_type = c("difference", "ratio"),
  ...
)
```

Arguments

`x`	an object output from `beaver_mcmc()`.
`doses`	doses at which to obtain the posterior.
`reference_dose`	dose to which to compare as either a difference or ratio.
`prob`	the percentiles of the posterior to calculate for each dose.
`return_stats`	logical indicating if the posterior mean and quantiles should be returned.
`return_samples`	logical indicating if posterior mean samples should be returned.
`new_data`	a dataframe for which the posterior will be calculated for each observation's covariate values.
`contrast`	a matrix containing where each row contains a contrast for which the posterior will be calculated.
`reference_type`	whether to provide the posterior of the difference or the ratio between each dose and the reference dose.
`...`	additional arguments will throw an error.

Value

A list with the elements `stats` and `samples`. When using this function with default settings, `samples` is NULL and `stats` is a dataframe summarizing the posterior samples. `stats` contains, at a minimum, the columns "dose", ".contrast_index", "(Intercept)", "value", and variables corresponding to the values passed in `prob` ("2.50%" and "97.50%" by default). When `return_stats` is set to `FALSE`, `stats` is NULL. When `return_samples` is set to `TRUE`, `samples` is a dataframe with the posterior samples for each iteration of the MCMC. The dataframe will have, at a minimum, the columns "iter" and "model", indicating the MCMC iteration and the model that was used in the calculations, as well as the columns "dose", ".contrast_index", "(Intercept)", and "value". The functions used for each model are defined within the `model_negbin_XYZ()` functions and used in the `beaver_mcmc()` function.

See Also

Other posterior calculations: `beaver_mcmc()`, `posterior.beaver_mcmc()`, `posterior_g_comp()`, `pr_eoi_g_comp()`, `pr_eoi()`

Examples

```
# The {beaver} package, by definition, performs MCMC for multiple models.
# Even with a small number of chains/burn-ins/samples, a minimally illustrative
# example requires >5s to run.

library(dplyr)

# No covariates----

set.seed(100)

df <- data_negbin_emax(
  n_per_arm = 10,
  doses = 0:3,
  b1 = 0,
  b2 = 2.5,
  b3 = 0.5,
  ps = 0.75
)

df %>%
  group_by(dose) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
```

```
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    linear = model_negbin_linear(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      w_prior = 1 / 4
    ),
    quad = model_negbin_quad(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    exp = model_negbin_exp(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 0,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    formula = ~ 1,
    data = df,
    n_iter = 1e2,
    n_chains = 1,
    quiet = TRUE
  )

  mcmc$w_post

  draws <- try(draws(mcmc)) #draws() is intended for single model fits only
  draws_emax <- draws(mcmc$models$emax$mcmc)
  draws_linear <- draws(mcmc$models$linear$mcmc)
  draws_quad <- draws(mcmc$models$quad$mcmc)
  draws_exp <- draws(mcmc$models$exp$mcmc)

  post <- posterior(
    mcmc,
    contrast = matrix(1, 1, 1),
    doses = 0:3,
    reference_dose = 0,
    reference_type = "difference"
  )
```

```
pr_eoi(
  mcmc,
  eoi = c(5, 8),
  contrast = matrix(1, 1, 1),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp <- posterior_g_comp(
  mcmc,
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc,
  eoi = c(5, 8),
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc, contrast = matrix(1, 1, 1))

# With covariates----

set.seed(1000)

x <-
  data.frame(
    gender = factor(sample(c("F", "M"), 40, replace = TRUE))
  ) %>%
  model.matrix(~ gender, data = .)

df_cov <-
  data_negbin_emax(
    n_per_arm = 10,
    doses = 0:3,
    b1 = c(0, 0.5),
    b2 = 2.5,
    b3 = 0.5,
    ps = 0.75,
    x = x
  ) %>%
  mutate(
    gender = case_when(
      genderM == 1 ~ "M",
      TRUE ~ "F"
    ),
    gender = factor(gender)
  ) %>%
```

```
  select(subject, dose, gender, response)

df_cov %>%
  group_by(dose, gender) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc_cov <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  linear = model_negbin_linear(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    w_prior = 1 / 4
  ),
  quad = model_negbin_quad(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ gender,
  data = df_cov,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc_cov$w_post
```

```
draws_cov <- try(draws(mcmc_cov)) #draws() is intended for single model fits only
draws_cov_emax <- draws(mcmc_cov$models$emax$mcmc)
draws_cov_linear <- draws(mcmc_cov$models$linear$mcmc)
draws_cov_quad <- draws(mcmc_cov$models$quad$mcmc)
draws_cov_exp <- draws(mcmc_cov$models$exp$mcmc)

post_cov <- posterior(
  mcmc_cov,
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc_cov,
  eoi = c(5, 8),
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp_cov <- posterior_g_comp(
  mcmc_cov,
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc_cov,
  eoi = c(5, 8),
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc_cov, new_data = df_cov, type = "g-comp")
```

`posterior_g_comp` *Compute Posterior G-Computation Estimate*

Description

Calculate the estimated effect for each observation at each dose and average over all observations. This function calculates the posterior marginal treatment effect at each dose.

Usage

```
posterior_g_comp(
  x,
  doses = attr(x, "doses"),
  reference_dose = NULL,
  prob = c(0.025, 0.975),
  return_stats = TRUE,
  return_samples = FALSE,
  new_data = NULL,
  reference_type = c("difference", "ratio")
)
```

Arguments

`x`	an object output from `beaver_mcmc()` or (internal function) `run_mcmc()`.
`doses`	doses at which to obtain the posterior.
`reference_dose`	dose to which to compare as either a difference or ratio.
`prob`	the percentiles of the posterior to calculate for each dose.
`return_stats`	logical indicating if the posterior mean and quantiles should be returned.
`return_samples`	logical indicating if posterior mean samples should be returned.
`new_data`	a dataframe containing all the variables used in the covariate adjustments to the model used to obtain `x`. Usually this will be the same dataframe used to fit the model.
`reference_type`	whether to provide the posterior of the difference or the ratio between each dose and the reference dose.

Value

A list with the elements `stats` and `samples`. When using this function with default settings, `samples` is NULL and `stats` is a dataframe summarizing the posterior samples. `stats` contains, at a minimum, the columns "dose", "value", and variables corresponding to the values passed in `prob` ("2.50%" and "97.50%" by default). When `return_stats` is set to `FALSE`, `stats` is NULL. When `return_samples` is set to `TRUE`, `samples` is a dataframe with the posterior samples for each iteration of the MCMC.

When x is of class 'beaver_mcmc_bma': The dataframe will have, at a minimum, the columns "iter" and "model", indicating the MCMC iteration and the model that was used in the calculations, as well as the columns "dose" and "value". The functions used for each model are defined within the `model_negbin_XYZ()` functions and used in the `beaver_mcmc()` function.

When x is of class 'beaver_mcmc': The dataframe will have, at a minimum, the column "iter", indicating the MCMC iteration, as well as the columns "dose" and "value". The functions used for each model are defined within the `model_negbin_XYZ()` functions and used in the `run_mcmc()` function.

See Also

Other posterior calculations: `beaver_mcmc()`, `posterior.beaver_mcmc_bma()`, `posterior.beaver_mcmc()`, `pr_eoi_g_comp()`, `pr_eoi()`

Examples

```
# The {beaver} package, by definition, performs MCMC for multiple models.
# Even with a small number of chains/burn-ins/samples, a minimally illustrative
# example requires >5s to run.

library(dplyr)

# No covariates----

set.seed(100)

df <- data_negbin_emax(
  n_per_arm = 10,
  doses = 0:3,
  b1 = 0,
  b2 = 2.5,
  b3 = 0.5,
  ps = 0.75
)

df %>%
  group_by(dose) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  linear = model_negbin_linear(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    w_prior = 1 / 4
  ),
  quad = model_negbin_quad(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
```

```
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ 1,
  data = df,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc$w_post

draws <- try(draws(mcmc)) #draws() is intended for single model fits only
draws_emax <- draws(mcmc$models$emax$mcmc)
draws_linear <- draws(mcmc$models$linear$mcmc)
draws_quad <- draws(mcmc$models$quad$mcmc)
draws_exp <- draws(mcmc$models$exp$mcmc)

post <- posterior(
  mcmc,
  contrast = matrix(1, 1, 1),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc,
  eoi = c(5, 8),
  contrast = matrix(1, 1, 1),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp <- posterior_g_comp(
  mcmc,
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc,
  eoi = c(5, 8),
```

```
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc, contrast = matrix(1, 1, 1))

# With covariates----

set.seed(1000)

x <-
  data.frame(
    gender = factor(sample(c("F", "M"), 40, replace = TRUE))
  ) %>%
  model.matrix(~ gender, data = .)

df_cov <-
  data_negbin_emax(
    n_per_arm = 10,
    doses = 0:3,
    b1 = c(0, 0.5),
    b2 = 2.5,
    b3 = 0.5,
    ps = 0.75,
    x = x
  ) %>%
  mutate(
    gender = case_when(
      genderM == 1 ~ "M",
      TRUE ~ "F"
    ),
    gender = factor(gender)
  ) %>%
  select(subject, dose, gender, response)

df_cov %>%
  group_by(dose, gender) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc_cov <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
```

```
    ),
    linear = model_negbin_linear(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      w_prior = 1 / 4
    ),
    quad = model_negbin_quad(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    exp = model_negbin_exp(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 0,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    formula = ~ gender,
    data = df_cov,
    n_iter = 1e2,
    n_chains = 1,
    quiet = TRUE
  )

  mcmc_cov$w_post

  draws_cov <- try(draws(mcmc_cov)) #draws() is intended for single model fits only
  draws_cov_emax <- draws(mcmc_cov$models$emax$mcmc)
  draws_cov_linear <- draws(mcmc_cov$models$linear$mcmc)
  draws_cov_quad <- draws(mcmc_cov$models$quad$mcmc)
  draws_cov_exp <- draws(mcmc_cov$models$exp$mcmc)

  post_cov <- posterior(
    mcmc_cov,
    contrast = matrix(c(1, 1, 0, 1), 2, 2),
    doses = 0:3,
    reference_dose = 0,
    reference_type = "difference"
  )

  pr_eoi(
    mcmc_cov,
    eoi = c(5, 8),
    contrast = matrix(c(1, 1, 0, 1), 2, 2),
```

```
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp_cov <- posterior_g_comp(
  mcmc_cov,
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc_cov,
  eoi = c(5, 8),
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc_cov, new_data = df_cov, type = "g-comp")
```

`pr_eoi` *Calculate Probability of Meeting Effect of Interest*

Description

Calculate a posterior quantity such as Pr(trt_arm1 - trt_arm2 > eoi)

Usage

```
pr_eoi(
  x,
  eoi,
  doses = attr(x, "doses"),
  reference_dose = NULL,
  new_data = NULL,
  contrast = NULL,
  reference_type = c("difference", "ratio"),
  direction = c("greater", "less")
)
```

Arguments

`x`	an object output from `beaver_mcmc()` or (internal function) `run_mcmc()`.
`eoi`	effects of interest in the probability equation.
`doses`	doses at which to obtain the posterior.
`reference_dose`	dose to which to compare as either a difference or ratio.

`new_data`	a dataframe for which the posterior will be calculated for each observation's covariate values.
`contrast`	a matrix containing where each row contains a contrast for which the posterior will be calculated.
`reference_type`	whether to provide the posterior of the difference or the ratio between each dose and the reference dose.
`direction`	calculate whether the posterior quantity is greater or less than the eoi

Value

A dataframe or tibble with the posterior quantities.

See Also

Other posterior calculations: `beaver_mcmc()`, `posterior.beaver_mcmc_bma()`, `posterior.beaver_mcmc()`, `posterior_g_comp()`, `pr_eoi_g_comp()`

Examples

```
# The {beaver} package, by definition, performs MCMC for multiple models.
# Even with a small number of chains/burn-ins/samples, a minimally illustrative
# example requires >5s to run.

library(dplyr)

# No covariates----

set.seed(100)

df <- data_negbin_emax(
  n_per_arm = 10,
  doses = 0:3,
  b1 = 0,
  b2 = 2.5,
  b3 = 0.5,
  ps = 0.75
)

df %>%
  group_by(dose) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
```

```
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  linear = model_negbin_linear(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    w_prior = 1 / 4
  ),
  quad = model_negbin_quad(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ 1,
  data = df,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc$w_post

draws <- try(draws(mcmc)) #draws() is intended for single model fits only
draws_emax <- draws(mcmc$models$emax$mcmc)
draws_linear <- draws(mcmc$models$linear$mcmc)
draws_quad <- draws(mcmc$models$quad$mcmc)
draws_exp <- draws(mcmc$models$exp$mcmc)

post <- posterior(
  mcmc,
  contrast = matrix(1, 1, 1),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)
```

```
pr_eoi(
  mcmc,
  eoi = c(5, 8),
  contrast = matrix(1, 1, 1),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp <- posterior_g_comp(
  mcmc,
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc,
  eoi = c(5, 8),
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc, contrast = matrix(1, 1, 1))

# With covariates----

set.seed(1000)

x <-
  data.frame(
    gender = factor(sample(c("F", "M"), 40, replace = TRUE))
  ) %>%
  model.matrix(~ gender, data = .)

df_cov <-
  data_negbin_emax(
    n_per_arm = 10,
    doses = 0:3,
    b1 = c(0, 0.5),
    b2 = 2.5,
    b3 = 0.5,
    ps = 0.75,
    x = x
  ) %>%
  mutate(
    gender = case_when(
      genderM == 1 ~ "M",
      TRUE ~ "F"
    ),
    gender = factor(gender)
  ) %>%
```

```
  select(subject, dose, gender, response)

df_cov %>%
  group_by(dose, gender) %>%
  summarize(
    mean = mean(response),
    se = sd(response) / sqrt(n()),
    .groups = "drop"
  )

mcmc_cov <- beaver_mcmc(
  emax = model_negbin_emax(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  linear = model_negbin_linear(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    w_prior = 1 / 4
  ),
  quad = model_negbin_quad(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 1.5,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ gender,
  data = df_cov,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc_cov$w_post
```

```
draws_cov <- try(draws(mcmc_cov)) #draws() is intended for single model fits only
draws_cov_emax <- draws(mcmc_cov$models$emax$mcmc)
draws_cov_linear <- draws(mcmc_cov$models$linear$mcmc)
draws_cov_quad <- draws(mcmc_cov$models$quad$mcmc)
draws_cov_exp <- draws(mcmc_cov$models$exp$mcmc)

post_cov <- posterior(
  mcmc_cov,
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc_cov,
  eoi = c(5, 8),
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp_cov <- posterior_g_comp(
  mcmc_cov,
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc_cov,
  eoi = c(5, 8),
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc_cov, new_data = df_cov, type = "g-comp")
```

`pr_eoi_g_comp` *Calculate Probability of Meeting Effect of Interest using G-Computation*

Description

Calculate a posterior quantity such as Pr(trt_arm1 - trt_arm2 > eoi) based on the posterior marginal treatment effect at each dose.

Usage

```
pr_eoi_g_comp(
  x,
  eoi,
  doses = attr(x, "doses"),
  reference_dose = NULL,
  new_data = NULL,
  reference_type = c("difference", "ratio"),
  direction = c("greater", "less")
)
```

Arguments

`x`	an object output from `beaver_mcmc()` or (internal function) `run_mcmc()`.
`eoi`	effects of interest in the probability equation.
`doses`	doses at which to obtain the posterior.
`reference_dose`	dose to which to compare as either a difference or ratio.
`new_data`	a dataframe containing all the variables used in the covariate adjustments to the model used to obtain `x`. Usually this will be the same dataframe used to fit the model.
`reference_type`	whether to provide the posterior of the difference or the ratio between each dose and the reference dose.
`direction`	calculate whether the posterior quantity is greater or less than the eoi

Value

A dataframe or tibble with the posterior quantities.

See Also

Other posterior calculations: `beaver_mcmc()`, `posterior.beaver_mcmc_bma()`, `posterior.beaver_mcmc()`, `posterior_g_comp()`, `pr_eoi()`

Examples

```
# The {beaver} package, by definition, performs MCMC for multiple models.
# Even with a small number of chains/burn-ins/samples, a minimally illustrative
# example requires >5s to run.

library(dplyr)

# No covariates----

set.seed(100)

df <- data_negbin_emax(
  n_per_arm = 10,
```

```
    doses = 0:3,
    b1 = 0,
    b2 = 2.5,
    b3 = 0.5,
    ps = 0.75
  )

  df %>%
    group_by(dose) %>%
    summarize(
      mean = mean(response),
      se = sd(response) / sqrt(n()),
      .groups = "drop"
    )

  mcmc <- beaver_mcmc(
    emax = model_negbin_emax(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    linear = model_negbin_linear(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      w_prior = 1 / 4
    ),
    quad = model_negbin_quad(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    exp = model_negbin_exp(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 0,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    formula = ~ 1,
    data = df,
    n_iter = 1e2,
```

```
  n_chains = 1,
  quiet = TRUE
)

mcmc$w_post

draws <- try(draws(mcmc)) #draws() is intended for single model fits only
draws_emax <- draws(mcmc$models$emax$mcmc)
draws_linear <- draws(mcmc$models$linear$mcmc)
draws_quad <- draws(mcmc$models$quad$mcmc)
draws_exp <- draws(mcmc$models$exp$mcmc)

post <- posterior(
  mcmc,
  contrast = matrix(1, 1, 1),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc,
  eoi = c(5, 8),
  contrast = matrix(1, 1, 1),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp <- posterior_g_comp(
  mcmc,
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc,
  eoi = c(5, 8),
  new_data = df,
  reference_dose = 0,
  reference_type = "difference"
)

plot(mcmc, contrast = matrix(1, 1, 1))

# With covariates----

set.seed(1000)

x <-
  data.frame(
    gender = factor(sample(c("F", "M"), 40, replace = TRUE))
  ) %>%
```

```
    model.matrix(~ gender, data = .)

  df_cov <-
    data_negbin_emax(
      n_per_arm = 10,
      doses = 0:3,
      b1 = c(0, 0.5),
      b2 = 2.5,
      b3 = 0.5,
      ps = 0.75,
      x = x
    ) %>%
    mutate(
      gender = case_when(
        genderM == 1 ~ "M",
        TRUE ~ "F"
      ),
      gender = factor(gender)
    ) %>%
    select(subject, dose, gender, response)

  df_cov %>%
    group_by(dose, gender) %>%
    summarize(
      mean = mean(response),
      se = sd(response) / sqrt(n()),
      .groups = "drop"
    )

  mcmc_cov <- beaver_mcmc(
    emax = model_negbin_emax(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
      w_prior = 1 / 4
    ),
    linear = model_negbin_linear(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      w_prior = 1 / 4
    ),
    quad = model_negbin_quad(
      mu_b1 = 0,
      sigma_b1 = 10,
      mu_b2 = 0,
      sigma_b2 = 10,
      mu_b3 = 1.5,
      sigma_b3 = 3,
```

```
    w_prior = 1 / 4
  ),
  exp = model_negbin_exp(
    mu_b1 = 0,
    sigma_b1 = 10,
    mu_b2 = 0,
    sigma_b2 = 10,
    mu_b3 = 0,
    sigma_b3 = 3,
    w_prior = 1 / 4
  ),
  formula = ~ gender,
  data = df_cov,
  n_iter = 1e2,
  n_chains = 1,
  quiet = TRUE
)

mcmc_cov$w_post

draws_cov <- try(draws(mcmc_cov)) #draws() is intended for single model fits only
draws_cov_emax <- draws(mcmc_cov$models$emax$mcmc)
draws_cov_linear <- draws(mcmc_cov$models$linear$mcmc)
draws_cov_quad <- draws(mcmc_cov$models$quad$mcmc)
draws_cov_exp <- draws(mcmc_cov$models$exp$mcmc)

post_cov <- posterior(
  mcmc_cov,
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  doses = 0:3,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi(
  mcmc_cov,
  eoi = c(5, 8),
  contrast = matrix(c(1, 1, 0, 1), 2, 2),
  reference_dose = 0,
  reference_type = "difference"
)

post_g_comp_cov <- posterior_g_comp(
  mcmc_cov,
  new_data = df_cov,
  reference_dose = 0,
  reference_type = "difference"
)

pr_eoi_g_comp(
  mcmc_cov,
  eoi = c(5, 8),
  new_data = df_cov,
```

```
    reference_dose = 0,
    reference_type = "difference"
  )

  plot(mcmc_cov, new_data = df_cov, type = "g-comp")
```

Index