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  ................................................................. 2
  data_negbin_emax  ......................................................... 7
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.
beaver_mcmc

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,
```
beaver_mcmc

    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
)
data_negbin_emax

Generate data from a negative binomial EMAX model

Description

Generate data from a negative binomial EMAX model
data_negbin_emax

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 co-
variants 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 \( j \)th subject on dose \( d_i \). The model is

\[
\begin{align*}
  y_{ij} &\sim \text{NB}(p_i, r_i) \\
p_i &\sim \text{Uniform}(0, 1) \\
r_{ij} &= (\mu_{ij} * p_i) / (1 - p_i) \\
\log(\mu_{ij}) &= x_{ij} * b1 + b2 * d_i / (b3 + d_i) \\
  b1 &\sim N(\mu_{b1}, \sigma_{b1}^2) \\
  b2 &\sim N(\mu_{b2}, \sigma_{b2}^2) \\
  b3 &\sim N(\mu_{b3}, \sigma_{b3}^2) (\text{Truncated to be positive})
\end{align*}
\]

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,
data_negbin_emax

\[
\begin{align*}
\text{mu}_b1 &= 0, \\
\text{sigma}_b1 &= 10, \\
\text{mu}_b2 &= 0, \\
\text{sigma}_b2 &= 10, \\
\text{mu}_b3 &= 1.5, \\
\text{sigma}_b3 &= 3, \\
w_{\text{prior}} &= 1 / 4 \\
\end{align*}
\]

\[
\begin{align*}
\text{mu}_b1 &= 0, \\
\text{sigma}_b1 &= 10, \\
\text{mu}_b2 &= 0, \\
\text{sigma}_b2 &= 10, \\
\text{mu}_b3 &= 0, \\
\text{sigma}_b3 &= 3, \\
w_{\text{prior}} &= 1 / 4 \\
\end{align*}
\]

\[
\begin{align*}
\text{formula} &= \sim \text{gender}, \\
data &= \text{df_cov}, \\
n_{\text{iter}} &= 1\text{e}2, \\
n_{\text{chains}} &= 1, \\
\text{quiet} &= \text{TRUE}
\end{align*}
\]

mcmc_cov$\text{w_post}

draws_cov <- try(draws(mcmc_cov)) #\text{draws()} is intended for single model fits only
draws_cov\text{_emax} <- draws(mcmc_cov$models$emax$mcmc)
draws_cov\text{_linear} <- draws(mcmc_cov$models$linear$mcmc)
draws_cov\text{_quad} <- draws(mcmc_cov$models$quad$mcmc)
draws_cov\text{_exp} <- draws(mcmc_cov$models$exp$mcmc)

\[
\begin{align*}
\text{contrast} &= \text{matrix}(c(1, 1, 0, 1), 2, 2), \\
doses &= 0:3, \\
\text{reference_dose} &= 0, \\
\text{reference_type} &= \text{"difference"}
\end{align*}
\]

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"
)
```
draws
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, ...)
```

**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----

```r
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,
```r
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**

```r
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 $j$th subject on dose $d_i$. The model is

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

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

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

$$\log(\mu_{ij}) = x_{ij} * b_1 + b_2 * d_i / (b_3 + d_i)$$

$$b_1 \sim N(\mu_{b1}, \sigma_{b1}^2)$$

$$b_2 \sim N(\mu_{b2}, \sigma_{b2}^2)$$

$$b_3 \sim N(\mu_{b3}, \sigma_{b3}^2) (Truncated to be positive)$$

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**

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

\texttt{mu\_b1, sigma\_b1, mu\_b2, sigma\_b2, mu\_b3, sigma\_b3}

hyperparameters. See the model description below for context.

\texttt{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 \(j\)th subject on dose \(d_i\). The model is

\[
y_{ij} \sim \text{NB}(p_i, r_{ij})
\]

\(p_i \sim \text{Uniform}(0, 1)\)

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

\[
\log(\mu_{ij}) = x_{ij} \ast b1 + b2 \ast (1 - \exp(-b3 \ast d_i))
\]

\(b1 \sim \text{N}(\text{`mu\_b1'}, \text{`sigma\_b1'^2})\)

\(b2 \sim \text{N}(\text{`mu\_b2'}, \text{`sigma\_b2'^2})\)

\(b3 \sim \text{N}(\text{`mu\_b3'}, \text{`sigma\_b3'^2})(\text{Truncated to positive})\)

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: \texttt{beaver\_mcmc()}, \texttt{model\_negbin\_emax()}, \texttt{model\_negbin\_indep()}, \texttt{model\_negbin\_linear()}, \texttt{model\_negbin\_loglinear()}, \texttt{model\_negbin\_logquad()}, \texttt{model\_negbin\_quad()}, \texttt{model\_negbin\_sigmoid\_emax()}
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**

```r
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 \( j \)th subject on the \( k \)th dose. The model is

\[
y_{ij} \sim NB(p_i, r_i)
\]

- \( p_i \sim Uniform(0, 1) \)
- \( r_{ij} = (\mu_{ij} \times p_i)/(1 - p_i) \)
- \( \log(\mu_{ij}) = x_{ij} \times b_1 + b_{2k} \)
- \( b_1 \sim N(\text{'mu_b1'}, \text{'sigma_b1'}^2) \)
- \( b_{2k} \sim N(\text{'mu_b2'}, \text{'sigma_b2'}^2) \)

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()`
Description

Model settings for a negative binomial distribution assuming a 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 \( j \)th subject on dose \( d_i \). The model is

\[
y_{ij} \sim NB(p_i, r_i) \\
p_i \sim Uniform(0, 1) \\
r_{ij} = (\mu_{ij} * p_i)/(1 - p_i) \\
\log(\mu_{ij}) = x_{ij} * b1 + b2 * d_i \\
b1 \sim N('mu_b1', 'sigma_b1^2') \\
b2 \sim N('mu_b2', 'sigma_b2^2')
\]

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 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()
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 $j$th subject on dose $d_i$. The model is

\[
y_{ij} \sim NB(p_i, r_{ij})
\]

\[
p_i \sim Uniform(0, 1)
\]

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

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

\[
b1 \sim Normal(\mu_{b1}, \sigma_{b1}^2)
\]

\[
b2 \sim Normal(\mu_{b2}, \sigma_{b2}^2)
\]

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()
Description

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

Usage

```r
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 $j$th subject on dose $d_i$. The model is

$$
y_{ij} \sim NB(p_i, r_{ij})
$$

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

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

$$
log(\mu_{ij}) = x_{ij} \cdot b_1 + b_2 \cdot log(1 + d_i) + b_3 \cdot log(1 + d_i)^2
$$

$$
b_1 \sim N('mu_b1', 'sigma_b1^2')
$$

$$
b_2 \sim N('mu_b2', 'sigma_b2^2')
$$

$$
b_3 \sim N('mu_b3', 'sigma_b3^2')
$$

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**

```r
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 $j$th subject on dose $d_i$. The model is

$$y_{ij} \sim NB(p_i, r_{ij})$$

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

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

$$log(\mu_{ij}) = x_{ij} * b_1 + b_2 * d_i + b3 * d_i^2$$

$$b_1 \sim N(\mu_{b1}, \sigma_{b1}^2)$$

$$b_2 \sim N(\mu_{b2}, \sigma_{b2}^2)$$
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 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()`
Negative Binomial Sigmoidal EMAX

Let $y_{ij}$ be the $j$th subject on dose $d_i$. The model is

$$y_{ij} \sim NB(p_i, r_{ij})$$

$$p_i \sim \text{Uniform}(0, 1)$$

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

$$\log(\mu_{ij}) = x_{ij} * b_1 + b_2 * d_i^4 / (b_3 + d_i^4)$$

$$b_1 \sim N('mu_{b1}', 'sigma_{b1}^2')$$

$$b_2 \sim N('mu_{b2}', 'sigma_{b2}^2')$$

$$b_3 \sim N('mu_{b3}', 'sigma_{b3}^2') \text{(Truncated to be positive)}$$

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 $\text{Uniform}(0, 1)$. 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

```r
## 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,
posterior.beaver_mcmc

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,
\[
\begin{align*}
\text{w\_prior} & = 1 / 4 \\
\exp & = \text{model\_negbin\_exp(} \\
& \quad \begin{align*}
\text{mu\_b1} & = 0, \\
\text{sigma\_b1} & = 10, \\
\text{mu\_b2} & = 0, \\
\text{sigma\_b2} & = 10, \\
\text{mu\_b3} & = 0, \\
\text{sigma\_b3} & = 3, \\
\text{w\_prior} & = 1 / 4 \\
\end{align*} \\
) \\
\text{formula} & = \sim \text{gender}, \\
\text{data} & = \text{df\_cov}, \\
\text{n\_iter} & = 1e2, \\
\text{n\_chains} & = 1, \\
\text{quiet} & = \text{TRUE} \\
\end{align*}
\]

\text{mcmc\_cov}\$w\_post

\text{draws\_cov} <- \text{try(draws(mcmc\_cov))} \# \text{draws()} is intended for single model fits only
\text{draws\_cov\_emax} <- \text{draws(mcmc\_cov\$models\$emax\$mcmc)}
\text{draws\_cov\_linear} <- \text{draws(mcmc\_cov\$models\$linear\$mcmc)}
\text{draws\_cov\_quad} <- \text{draws(mcmc\_cov\$models\$quad\$mcmc)}
\text{draws\_cov\_exp} <- \text{draws(mcmc\_cov\$models\$exp\$mcmc)}

\text{post\_cov} <- \text{posterior(}
\text{mcmc\_cov},
\text{contrast} = \text{matrix(c(1, 1, 0, 1), 2, 2)},
\text{doses} = 0:3,
\text{reference\_dose} = 0,
\text{reference\_type} = \text{"difference"}
\text{)}

\text{pr\_eoi(}
\text{mcmc\_cov},
\text{eoi} = \text{c(5, 8)},
\text{contrast} = \text{matrix(c(1, 1, 0, 1), 2, 2)},
\text{reference\_dose} = 0,
\text{reference\_type} = \text{"difference"}
\text{)}

\text{post\_g\_comp\_cov} <- \text{posterior\_g\_comp(}
\text{mcmc\_cov},
\text{new\_data} = \text{df\_cov},
\text{reference\_dose} = 0,
\text{reference\_type} = \text{"difference"}
\text{)}

\text{pr\_eoi\_g\_comp(}
\text{mcmc\_cov},
\text{eoi} = \text{c(5, 8)},
\text{new\_data} = \text{df\_cov},
posterior.beaver_mcmc_bma

```r
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**

```r
# 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`: 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.
The `posterior.beaver_mcmc_bma` function returns a list with two elements: `stats` and `samples`.

- `stats`: A dataframe summarizing the posterior samples. At a minimum, it contains 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.

- `samples`: A dataframe with the posterior samples for each iteration of the MCMC. At a minimum, it contains the columns "iter", "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

```r
# 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,
  ),
  .prob = c(2.5, 97.5),
  .iter = 500,
  .burn_in = 100,
  .samples = 1000,
  .return_stats = TRUE,
  .return_samples = TRUE
)
```


\begin{verbatim}

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"
)
\end{verbatim}
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)
)  # %>
posterior.beaver_mcmc_bma

```r
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$w_post
```
posterior_g_comp <- try(draws(mcmc_cov)) #draws() is intended for single model fits only
draws_cov_emax <- draws(mcmc_cov$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

```r
code
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,
posterior_g_comp

\[
\begin{align*}
&\sigma_b3 = 3, \\
&w_{prior} = 1/4
\end{align*}
\]

\[
exp = \text{model\_negbin\_exp}(
\begin{align*}
&m_u_{b1} = 0, \\
&\sigma_{b1} = 10, \\
&m_u_{b2} = 0, \\
&\sigma_{b2} = 10, \\
&m_u_{b3} = 0, \\
&\sigma_{b3} = 3, \\
&w_{prior} = 1/4
\end{align*}
\),
\]

\[
\text{formula} = -1, \\
\text{data} = \text{df}, \\
n_{iter} = 1e2, \\
n_{chains} = 1, \\
\text{quiet} = \text{TRUE}
\]

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
  )
)

posterior_g_comp

),
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,
eti = c(5, 8),
    contrast = matrix(c(1, 1, 0, 1), 2, 2),
pr_eoi <- posterior_g_comp(
  mcmc_cov,
  new_data = df_cov,
  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(tmt_arm1 - tmt_arm2 > eoi)

**Usage**

```r
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.
pr_eoi

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$m_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

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 $\Pr(\text{trt\_arm1} - \text{trt\_arm2} > \text{eoi})$ 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 = \frac{1}{4}
),
exp = \text{model\_negbin\_exp}(
  \mu_{b1} = 0,
  \sigma_{b1} = 10,
  \mu_{b2} = 0,
  \sigma_{b2} = 10,
  \mu_{b3} = 0,
  \sigma_{b3} = 3,
  w_{\text{prior}} = \frac{1}{4}
),
formula = \sim \text{gender},
data = \text{df\_cov},
n_iter = 1e2,
n_chains = 1,
quiet = \text{TRUE}
)

\text{mcmc\_cov}\$w\_post

draws\_cov <- \text{try(draws(mcmc\_cov))} \quad \# \text{draws()} \text{ is intended for single model fits only}
draws\_cov\_emax <- \text{draws(mcmc\_cov}\$\text{models}\$\text{emax}\$\text{mcmc})
draws\_cov\_linear <- \text{draws(mcmc\_cov}\$\text{models}\$\text{linear}\$\text{mcmc})
draws\_cov\_quad <- \text{draws(mcmc\_cov}\$\text{models}\$\text{quad}\$\text{mcmc})
draws\_cov\_exp <- \text{draws(mcmc\_cov}\$\text{models}\$\text{exp}\$\text{mcmc})

\text{post\_cov} \leftarrow \text{posterior}(
  \text{mcmc\_cov},
  \text{contrast} = \text{matrix(c(1, 1, 0, 1), 2, 2)},
  \text{doses} = 0:3,
  \text{reference\_dose} = 0,
  \text{reference\_type} = \text{"difference"}
)

\text{pr\_eoi}(
  \text{mcmc\_cov},
  \text{eoi} = c(5, 8),
  \text{contrast} = \text{matrix(c(1, 1, 0, 1), 2, 2)},
  \text{reference\_dose} = 0,
  \text{reference\_type} = \text{"difference"}
)

\text{post\_g\_comp\_cov} \leftarrow \text{posterior\_g\_comp}(
  \text{mcmc\_cov},
  \text{new\_data} = \text{df\_cov},
  \text{reference\_dose} = 0,
  \text{reference\_type} = \text{"difference"}
)

\text{pr\_eoi\_g\_comp}(
  \text{mcmc\_cov},
  \text{eoi} = c(5, 8),
  \text{new\_data} = \text{df\_cov},
reference_dose = 0,
reference_type = "difference"
)

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

* models
  beaver_mcmc, 2
  model_negbin_emax, 18
  model_negbin_exp, 19
  model_negbin_indep, 21
  model_negbin_linear, 22
  model_negbin_loglinear, 23
  model_negbin_logquad, 24
  model_negbin_quad, 25
  model_negbin_sigmoid_emax, 26

* posterior calculations
  beaver_mcmc, 2
  posterior.beaver_mcmc, 27
  posterior.beaver_mcmc_bma, 33
  posterior_g_comp, 38
  pr_eoi, 44
  pr_eoi_g_comp, 49

beaver_mcmc, 2, 19–23, 25–28, 34, 39, 45, 50

data_negbin_emax, 7
draws, 13

model_negbin_emax, 3, 18, 20–23, 25–27
model_negbin_exp, 3, 19, 19, 21–23, 25–27
model_negbin_indep, 3, 19, 20, 21, 22, 23, 25–27
model_negbin_linear, 3, 19–21, 22, 23, 25–27
model_negbin_loglinear, 3, 19–22, 23, 25–27
model_negbin_logquad, 3, 19–23, 24, 26, 27
model_negbin_quad, 3, 19–23, 25, 25, 27
model_negbin_sigmoid_emax, 3, 19–23, 25, 26, 26

posterior.beaver_mcmc, 3, 27, 34, 39, 45, 50
posterior.beaver_mcmc_bma, 3, 28, 33, 39, 45, 50
posterior_g_comp, 3, 28, 34, 38, 45, 50