Package ‘tipmap’

August 14, 2023

Type Package

Title Tipping Point Analysis for Bayesian Dynamic Borrowing

Version 0.5.2

Description Tipping point analysis for clinical trials that employ Bayesian dynamic borrowing via robust meta-analytic predictive (MAP) priors. Further functions facilitate expert elicitation of a primary weight of the informative component of the robust MAP prior and computation of operating characteristics. Intended use is the planning, analysis and interpretation of extrapolation studies in pediatric drug development, but applicability is generally wider.

License Apache License 2.0

URL https://github.com/Boehringer-Ingelheim/tipmap

BugReports https://github.com/Boehringer-Ingelheim/tipmap/issues

Encoding UTF-8

Imports dplyr, magrittr, purrr, ggplot2, RBesT, assertthat, stats, furrr, future

Suggests knitr, rmarkdown, tidyrr, tidble, testthat (>= 3.0.0),

Config/testthat/edition 3

RoxygenNote 7.2.3

VignetteBuilder knitr

Depends R (>= 3.5.0)

NeedsCompilation no

Repository CRAN

Author Christian Stock [aut, cre] (<https://orcid.org/0000-0002-3493-3234>), Morten Dreher [aut], Emma Torrini [ctb], Boehringer Ingelheim Pharma GmbH & Co. KG [cph, fnd]

Maintainer Christian Stock <christian.stock@boehringer-ingelheim.com>

Date/Publication 2023-08-14 10:30:03 UTC
**R topics documented:**

create_new_trial_data ........................................ 2
create_posterior_data ........................................ 3
create_prior_data ........................................... 4
create_tipmap_data ........................................... 5
default_quantiles ........................................... 6
default_weights ............................................... 6
draw_beta_mixture_nsamples ................................. 7
fit_beta_1exp .................................................. 8
fit_beta_mult_exp ............................................. 8
get_cum_probs_1exp .......................................... 9
get_model_input_1exp ....................................... 10
get_posterior_by_weight .................................... 11
get_summary_mult_exp ...................................... 11
get_tipping_points ......................................... 12
load_tipmap_data ........................................... 13
oc_bias .......................................................... 13
oc_coverage ................................................... 15
oc_pos .......................................................... 16
tipmap_darkblue ............................................. 18
tipmap_lightred ............................................... 18
tipmap_plot .................................................... 19

---

create_new_trial_data  
*Data on new trial in target population*

**Description**

Creates a vector containing data on the new trial in the target population. This may be hypothetical data in the planning stage.

**Usage**

create_new_trial_data(n_total, est, se)

**Arguments**

- **n_total**: The total sample size.
- **est**: Treatment effect estimate.
- **se**: Standard error of the treatment effect estimate.

**Value**

A numeric vector with data on the new trial, incl. quantiles of an assumed normal data likelihood.
create_posterior_data

See Also

create_posterior_data, create_tipmap_data

Examples

```r
new_trial_data <- create_new_trial_data(
  n_total = 30, est = 1.27, se = 0.95
)
```

create_posterior_data Creates posterior distributions for a range of weights on the informative component of the robust MAP prior

Description

Returns a data frame containing the default quantiles of posterior mixture distributions generated with varying weights on the informative component of the MAP prior.

Usage

```r
create_posterior_data(map_prior, new_trial_data, sigma, null_effect = 0)
```

Arguments

- `map_prior`: A MAP prior containing information about the trial(s) in the source population, created using RBesT.
- `new_trial_data`: A vector containing information about the new trial. See `create_new_trial_data()`.
- `sigma`: Standard deviation to be used for the weakly informative component of the MAP prior, recommended to be the unit-information standard deviation.
- `null_effect`: The mean of the robust component of the MAP prior. Defaults to 0.

Value

A data frame containing posterior distributions for varying weights

References


See Also

create_new_trial_data, create_prior_data
create_prior_data

Creates input data frame for construction of MAP prior

Description

Assembling information from trials in the source population in a structured way (required as a pre-processing step for MAP prior creation).

Usage

create_prior_data(study_label = NULL, n_total, est, se)

Arguments

study_label An optional vector containing trial labels.
n_total A vector containing total sample sizes.
est A vector containing treatment effect estimates.
se A vector containing standard errors of the effect estimates.

Value

A data frame containing data on the trials in the source population.

See Also

create_new_trial_data, create_posterior_data
create_tipmap_data

Examples

prior_data <- create_prior_data(
  n_total = c(160, 240, 320),
  est = c(1.23, 1.40, 1.51),
  se = c(0.4, 0.36, 0.31)
)

create_tipmap_data  Create data frame ready to use for tipping point analysis

Description

Combines new trial data created by createTargetData(), a posterior distribution created by create_posterior_data() and a robust MAP prior using RBesT::automixfit() and an optional meta-analysis, e.g. created using the meta package, into a data frame needed for the functions tipmap_plot() and get_tipping_point().

Usage

create_tipmap_data(new_trial_data, posterior, map_prior, meta_analysis = NULL)

Arguments

new_trial_data  A data frame containing data on the new trial in the target population. See create_new_trial_data().
posterior  A mixture combining MAP prior and target population. See create_posterior_data().
map_prior  A robust MAP prior created by RBesT::automixfit().
meta_analysis  A data frame containing a meta-analysis of trial(s) to be borrowed from. See createPriorData().

Value

A data frame ready to be used for tipmap_plot() and get_tipping_point()

See Also

create_new_trial_data, create_posterior_data, tipmap_plot, get_tipping_points

Examples

# specify new trial data
new_trial_data <- create_new_trial_data(n_total = 30, est = 1.5, se = 2.1)

# read MAP prior data
map_prior <- load_tipmap_data("tipmapPrior.rds")
# read posterior data
posterior <- load_tipmap_data("tipPost.rds")

tip_dat <- create_tipmap_data(
    new_trial_data = new_trial_data,
    posterior = posterior,
    map_prior = map_prior
)

---

### default_quantiles

**Default quantiles**

**Description**
Default quantiles

**Usage**
default_quantiles

**Format**
An object of class numeric of length 13.

---

### default_weights

**Default weights**

**Description**
Default weights

**Usage**
default_weights

**Format**
An object of class numeric of length 201.
**draw_beta_mixture_nsamples**

*Draw samples from a mixture of beta distributions*

**Description**

Draws samples from a mixture of beta distributions, representing pooled weights on the informative component of a robust MAP prior, as elicited from experts via the roulette method.

**Usage**

```r
draw_beta_mixture_nsamples(n, chips_mult, expert_weight = NULL)
```

**Arguments**

- `n` Numeric value, the number of samples to be drawn.
- `chips_mult` Numeric matrix, containing expert weighting (distributions of chips). Rows should represent experts, columns should represent bins / weight intervals.
- `expert_weight` An optional numeric vector, containing the weight assigned to each expert (defaults to equal weights).

**Value**

A numeric vector containing samples from a pooled distribution of expert opinions.

**See Also**

`fit_beta_mult_exp` and `get_summary_mult_exp`.

**Examples**

```r
rweights <- draw_beta_mixture_nsamples(
  n = 50,
  chips_mult = rbind(
    c(0, 0, 0, 2, 3, 2, 0, 0),
    c(0, 0, 1, 2, 4, 2, 1, 0),
    c(0, 0, 2, 2, 2, 2, 0, 0)
  ),
  expert_weight = rep(1/3, 3)
)

print(rweights)
## Not run:
hist(rweights)
## End(Not run)
```
fit_beta_1exp

Fit beta distribution for one expert

Description
Fit beta distribution to data elicited from one expert via the roulette method.

Usage
fit_beta_1exp(df)

Arguments
df
A dataframe generated by get_model_input_1exp.

Details
This function is based on SHELF::fitdist and yields identical results.

Value
Parameters (alpha and beta) of a beta fit.

See Also
get_model_input_1exp and fit_beta_mult_exp.

Examples
chips <- c(0, 2, 3, 2, 1, 1, 0, 0, 0)
x <- get_cum_probs_1exp(chips)
y <- get_model_input_1exp(x)
fit_beta_1exp(df = y)["par"]

fit_beta_mult_exp

Fit beta distributions for multiple experts

Description
Fit beta distributions to data elicited from multiple experts via the roulette method.

Usage
fit_beta_mult_exp(chips_mult)
Arguments

chips_mult  A dataframe or matrix containing weights. It should contain one row per expert and 10 columns, one for each bin, representing weights from 0 to 1.

Value

A dataframe containing the parameters of the individual beta distributions.

See Also

fit_beta_1exp.

Examples

beta_fits <- fit_beta_mult_exp(
  chips_mult = rbind(
    c(0, 0, 0, 2, 3, 3, 2, 0, 0),
    c(0, 0, 1, 2, 4, 2, 1, 0, 0),
    c(0, 0, 2, 2, 2, 2, 0, 0)
  )
)
print(beta_fits)
Examples

chips <- c(0, 2, 3, 2, 1, 1, 0, 0, 0, 0)
x <- get_cum_probs_1exp(chips)
print(x)

y <- get_model_input_1exp(x)
print(y)
**get_posterior_by_weight**

*Filter posterior by given weights*

**Description**

Returns quantiles of the posterior distribution of the treatment effect for one or more specified weights.

**Usage**

```r
get_posterior_by_weight(posterior, weight)
```

**Arguments**

- `posterior`: The posterior data to be filtered (see `create_posterior_data()`).
- `weight`: The weight(s) to be filtered by.

**Value**

The filtered posterior values

**See Also**

`create_posterior_data`

**Examples**

```r
get_posterior_by_weight(
  posterior = load_tipmap_data("tipPost.rds"),
  weight = c(0.05, 0.1)
)
```

**get_summary_mult_exp**

*Summarize expert weights*

**Description**

Computes minimum, maximum, mean and quartiles for expert weights.

**Usage**

```r
get_summary_mult_exp(chips_mult, n = 500, expert_weight = NULL)
```
get_tipping_points

Arguments

- **chips_mult**: Numeric matrix, containing expert weights.
- **n**: Number of samples to be drawn to obtain summary statistics (defaults to 500).
- **expert_weight**: Weights assigned to each expert (defaults to equal weights).

Value

A vector containing summary statistics.

Examples

```r
get_summary_mult_exp(
  chips_mult = rbind(
    c(0, 0, 0, 0, 2, 3, 3, 2, 0, 0),
    c(0, 0, 0, 1, 2, 4, 2, 1, 0, 0),
    c(0, 0, 0, 2, 2, 2, 2, 2, 0, 0)
  ),
  n = 50
)
```

get_tipping_points

Identify tipping point for a specific quantile.

Description

Identifies the weights closest to tipping points for specified quantiles.

Usage

```r
get_tipping_points(tipmap_data, quantile, null_effect = 0)
```

Arguments

- **tipmap_data**: A data frame created by `create_tipmap_data()`.
- **quantile**: The quantile(s) of the tipping point. Possible values are 0.025, 0.05, 0.1, 0.2, 0.8, 0.9, 0.95 and 0.975.
- **null_effect**: The null treatment effect. Defaults to 0.

Value

The weight closest to the tipping point for the specified quantile.

See Also

`create_tipmap_data`
load_tipmap_data

Examples

```r
tip_dat <- load_tipmap_data("tipdat.rds")#
get_tipping_points(tip_dat, quantile = 0.025)
get_tipping_points(tip_dat, quantile = c(0.025, 0.05, 0.1, 0.2), null_effect = 0.1)
```

Description

Loads one of three exemplary datasets in the package.

Usage

```r
load_tipmap_data(file)
```

Arguments

- `file` The dataset to be loaded.

Value

A pre-saved dataset.

Examples

```r
load_tipmap_data(file = "tipdat.rds")
load_tipmap_data(file = "tipmapPrior.rds")
load_tipmap_data(file = "tipPost.rds")
```

oc_bias

Assessing bias

Description

Assessment of absolute bias in posterior means and medians for a given weight and evidence level, using simulated data as input.
Usage

```r
oc_bias(
  m, 
  se, 
  true_effect, 
  weights = seq(0, 1, by = 0.01), 
  map_prior, 
  sigma, 
  n_cores = 1, 
  eval_strategy = "sequential"
)
```

Arguments

- **m** Numerical vector of simulated effect estimates.
- **se** Numerical vector of simulated standard errors (m and se need to have the same length).
- **true_effect** Numerical value, representing the true treatment effect (usually the mean of the simulated m).
- **weights** Vector of weights of the informative component of the MAP prior (defaults to seq(0, 1, by = 0.01)).
- **map_prior** A MAP prior containing information about the trials in the source population, created using RBesT; a mixture of normal distributions is required.
- **sigma** Standard deviation of the weakly informative component of the MAP prior, recommended to be the unit-information standard deviation.
- **n_cores** Integer value, representing the number of cores to be used (defaults to 1); only applies if eval_strategy is not "sequential".
- **eval_strategy** Character variable, representing the evaluation strategy, either "sequential", "multisession", or "multicore" (see documentation of future::plan, defaults to "sequential").

Value

A 2-dimensional array containing results on bias.

See Also

- `oc_pos` and `oc_coverage`.

Examples

```r
set.seed(123)
n_sims <- 5 # small number for exemplary application
sim_dat <- list(
  "m" = rnorm(n = n_sims, mean = 1.15, sd = 0.1),
  "se" = rnorm(n = n_sims, mean = 1.8, sd = 0.3)
)
```
Assessing coverage

Description

Assessment of coverage of posterior intervals for a given weight and evidence level, using simulated data as input.

Usage

```r
oc_coverage(
m, 
se, 
true_effect, 
weights = seq(0, 1, by = 0.01), 
map_prior, 
sigma, 
n_cores = 1, 
eval_strategy = "sequential"
)
```

Arguments

- `m` Numerical vector of simulated effect estimates.
- `se` Numerical vector of simulated standard errors (`m` and `se` need to have the same length).
- `true_effect` Numerical value, representing the true treatment effect (usually the mean of the simulated `m`).
- `weights` Vector of weights of the informative component of the MAP prior (defaults to `seq(0, 1, by = 0.01)`).
- `map_prior` A MAP prior containing information about the trials in the source population, created using RBesT; a mixture of normal distributions is required.
- `sigma` Standard deviation of the weakly informative component of the MAP prior, recommended to be the unit-information standard deviation.

```r
results <- oc_bias(
m = sim_dat["m"], 
se = sim_dat["se"], 
true_effect = 1.15, 
weights = seq(0, 1, by = 0.01), 
map_prior = load_tipmap_data("tipmapPrior.rds"), 
sigma = 16.23, 
eval_strategy = "sequential", 
n_cores = 1
)
print(results)
```
n_cores Integer value, representing the number of cores to be used (defaults to 1); only applies if `eval_strategy` is not "sequential".

eval_strategy Character variable, representing the evaluation strategy, either "sequential", "multisession", or "multicore" (see documentation of `future::plan`, defaults to "sequential").

Value

A 2-dimensional array containing results on coverage.

See Also

`oc_pos` and `oc_bias`.

Examples

```r
set.seed(123)
n_sims <- 5 # small number for exemplary application
sim_dat <- list(
  "m" = rnorm(n = n_sims, mean = 1.15, sd = 0.1),
  "se" = rnorm(n = n_sims, mean = 1.8, sd = 0.3)
)
results <- oc_coverage(
m = sim_dat[["m"]],
se = sim_dat[["se"]],
true_effect = 1.15,
weights = seq(0, 1, by = 0.01),
map_prior = load_tipmap_data("tipmapPrior.rds"),
sigma = 16.23
)
print(results)
```

Description

Assessment of the probability of truly or falsely (depending on simulated scenario) rejecting the null hypothesis of interest for a given weight and evidence level, using simulated data as input.

Usage

```r
oc_pos(
m, 
se, 
probs, 
weights = seq(0, 1, by = 0.01), 
map_prior,
```
Arguments

m  Numerical vector of simulated effect estimates.
se Numerical vector of simulated standard errors (m and se need to have the same length).
probs Vector of quantiles q, with 1 minus q representing an evidence level of interest (where positive effect estimate indicate a beneficial treatment).
weights Vector of weights of the informative component of the MAP prior (defaults to seq(0, 1, by = 0.01)).
map_prior A MAP prior containing information about the trials in the source population, created using RBesT; a mixture of normal distributions is required.
sigma Standard deviation of the weakly informative component of the MAP prior, recommended to be the unit-information standard deviation.
null_effect Numerical value, representing the null effect (defaults to 0).
direction_pos Logical value, TRUE (default) if effects greater that the null_effect indicate a beneficial treatment and FALSE otherwise.
n_cores Integer value, representing the number of cores to be used (defaults to 1); only applies if eval_strategy is not "sequential".
eval_strategy Character variable, representing the evaluation strategy, either "sequential", "multisession", or "multicore" (see documentation of future::plan, defaults to "sequential").

Value

A 2-dimensional array containing probabilities, either of truly (probability of success) or falsely rejecting the null hypothesis of interest for a given weight and evidence level.

See Also

oc_bias and oc_coverage.

Examples

set.seed(123)
n_sims <- 5 # small number for exemplary application
sim_dat <- list(
  "m" = rnorm(n = n_sims, mean = 1.15, sd = 0.1),
  "se" = rnorm(n = n_sims, mean = 1.8, sd = 0.3)
)
results <- oc_pos(
```r
m = sim_dat["m"],
se = sim_dat["se"],
probs = c(0.025, 0.05, 0.1, 0.2),
weights = seq(0, 1, by = 0.01),
map_prior = load_tipmap_data("tipmapPrior.rds"),
sigma = 16.23,
null_effect = 0,
direction_pos = TRUE,
eval_strategy = "sequential",
n_cores = 1
)
print(results)
```

---

### tipmap_darkblue

**Custom dark blue**

**Description**

Custom dark blue

**Usage**

```r
tipmap_darkblue
```

**Format**

An object of class character of length 1.

---

### tipmap_lightred

**Custom light red**

**Description**

Custom light red

**Usage**

```r
tipmap_lightred
```

**Format**

An object of class character of length 1.
tipmap_plot

Visualize tipping point analysis

Description

Uses a data frame created by create_tipmap_data() to visualize the tipping point analysis.

Usage

tipmap_plot(
  tipmap_data,
  target_pop_lab = "Trial in target\n population",
  y_range = NULL,
  y_breaks = NULL,
  title = NULL,
  y_lab = "Mean difference",
  x_lab = "Weight on informative component of MAP prior",
  map_prior_lab = "MAP\nprior",
  meta_analysis_lab = "MA",
  legend_title = "Posterior quantile",
  null_effect = 0
)

Arguments

tipmap_data  A data frame containing tipping point data, generated by create_tipmap_data().
target_pop_lab A label for the trial in the target population.
y_range An optional argument specifying range of the y-axis.
y_breaks An optional vector specifying breaks on the y-axis.
title The plot title.
y_lab The label for the y axis. Defaults to "Mean difference".
x_lab The label for the x axis. Defaults to "Weight on informative component of MAP prior".
map_prior_lab The label for the MAP prior. Defaults to "MAP prior"
meta_analysis_lab An optional label for a meta-analysis (if included).
legend_title An optional title for the plot legend. Defaults to "Posterior quantiles".
null_effect The null treatment effect, determining where tipping points are calculated. Defaults to 0.

Value

A ggplot object of the tipping point plot
See Also

`create_tipmap_data`

Examples

```r
timap_data <- load_tipmap_data("tipdat.rds")
timap_plot(timap_data)
```
Index

* datasets
  default_quantiles, 6
  default_weights, 6
  tipmap_darkblue, 18
  tipmap_lightred, 18

create_new_trial_data, 2, 3–5
create_posterior_data, 3, 3, 4, 5, 11
create_prior_data, 3, 4
create_tipmap_data, 3, 5, 12, 20

default_quantiles, 6
default_weights, 6
draw_beta_mixture_nsamples, 7

fit_beta_1exp, 8, 9, 10
fit_beta_mult_exp, 7, 8, 8

get_cum_probs_1exp, 9, 10
get_model_input_1exp, 8, 9, 10
get_posterior_by_weight, 11
get_summary_mult_exp, 7, 11
get_tipping_points, 5, 12

load_tipmap_data, 13

oc_bias, 13, 16, 17
oc_coverage, 14, 15, 17
oc_pos, 14, 16, 16

tipmap_darkblue, 18
tipmap_lightred, 18
tipmap_plot, 5, 19