Package ‘tipmap’

June 17, 2023

Type Package

Title Tipping Point Analysis for Bayesian Dynamic Borrowing

Version 0.4.2

Description Tipping point analysis for clinical trials that employ Bayesian dynamic borrowing via robust meta-analytic predictive (MAP) priors. Mainly an implementation of an approach proposed by Best and colleagues (2021) is provided <doi:10.1002/pst.2093>. Further functions facilitate the specification of the robust MAP prior via expert elicitation (using the roulette method). 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, purrr, ggplot2, RBesT

Suggests knitr, rmarkdown, tidyr, magrittr, tibble, testthat (>= 3.0.0),

Config/testthat/edition 3

RoxygenNote 7.2.3

VignetteBuilder knitr

Depends R (>= 3.5.0)

NeedsCompilation no

Repository CRAN

Date/Publication 2023-06-17 15:50:02 UTC

Author Morten Dreher [aut],

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

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

See Also

create_posterior_data, create_tipmap_data
create_posterior_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
default_weights = 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_1sample

Draw a single sample from a mixture of beta distributions

Description

Internal function needed for expert elicitation methods.

Usage

draw_beta_mixture_1sample(params, weights = NULL)

Arguments

params     Dataframe with parameters of beta distributions. Parameters need to be provided in columns named "alpha" and "beta", respectively, i.e. with one row per distribution.
weights    Optional vector of weights assigned to experts. Defaults to equal weights.

Value

One draw (numeric value) from a mixture of beta distributions.

Examples

beta_fits <- fit_beta_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)
   )
)
draw_beta_mixture_1sample(beta_fits)

draw_beta_mixture_nsamples

Draw n 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

draw_beta_mixture_nsamples(n, chips_mult, weights = NULL)

Arguments

n
  The number of samples to be drawn.

chips_mult
  A data frame or matrix containing expert weights. Rows should represent ex-
  perts, columns should represent bins / weights.

weights
  An optional 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.

Examples

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)
  )
)
rweights

---

fit_beta_1exp  Fit beta distribution for one expert

Description

Fit beta distribution to weights 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 SHELFF::fitdist and yields identical results.
Value

Parameters of beta fit.

Examples

chips <- c(0, 2, 3, 2, 1, 1, 0, 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(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

Parameters of the individual beta distributions.

Examples

beta_fits <- fit_beta_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)
  ))
beta_fits
get_cum_probs_1exp  Get cumulative probabilities from distribution of chips of one expert

Description
Internal function needed for expert elicitation methods.

Usage
get_cum_probs_1exp(chips)

Arguments
chips A numeric vector representing the distribution of chips of one expert. The vector must be of length 10 and contents must add up to 10. The first column represents weight on interval 0 to 0.1.

Value
A vector of cumulative probabilities.

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

get_model_input_1exp  Transform cumulative probabilities to fit beta distributions

Description
Internal function needed for expert elicitation methods.

Usage
get_model_input_1exp(cum_probs, w = NULL)

Arguments
cum_probs Cumulative probabilities of weights of one expert.
w Weight of bins.

Value
Dataframe that can be used as input to fit beta distributions by fit_beta_1exp().
Examples

chips <- c(0,2,3,2,1,1,1,0,0,0)
x <- get_cum_probs_1exp(chips)
y <- get_model_input_1exp(x)
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

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

get_posterior_by_weight(
  posterior = load_tquipmap_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, weights = NULL)
```

**Arguments**
- `chips_mult`: A data frame or matrix containing expert weights.
- `n`: The number of samples to be drawn to obtain summary statistics. Defaults to 500.
- `weights`: Weights assigned to each expert. Defaults to equal weights.

**Value**
A vector containing summary statistics.

**Examples**
```r
get_summary_mult_exp(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)
))
```

**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`

Examples

```r
load_tipmap_data(file = "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**

Custom dark blue

**Usage**

```r
tipmap_darkblue
```

**Format**

An object of class character of length 1.

---

**Description**

Custom light red

**Usage**

```r
tipmap_lightred
```

**Format**

An object of class character of length 1.

---

**Description**

Visualize tipping point analysis

**Usage**

```r
tipmap_plot
```

**Description**

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

```r
timap_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(tipmap_data)
```
Index

* datasets
  default_quantiles, 6
  default_weights, 6
  tipmap_darkblue, 14
  tipmap_lightred, 14

create_new_trial_data, 2, 3–5
create_posterior_data, 2, 3, 4, 5, 11
create_prior_data, 3, 4
create_tipmap_data, 2, 5, 13, 15

default_quantiles, 6
default_weights, 6
draw_beta_mixture_1sample, 7
draw_beta_mixture_nsamples, 7

fit_beta_1exp, 8
fit_beta_mult_exp, 9

get_cum_probs_1exp, 10
get_model_input_1exp, 10
get_posterior_by_weight, 11
get_summary_mult_exp, 12
get_tipping_points, 5, 12

load_tipmap_data, 13

tipmap_darkblue, 14
tipmap_lightred, 14
tipmap_plot, 5, 14