Package: tipmap (via r-universe)

September 1, 2024

Type Package

Title Tipping Point Analysis for Bayesian Dynamic Borrowing

Version 0.5.3

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, https://boehringer-ingelheim.github.io/tipmap/

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

Depends R (>= 3.5.0)

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

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

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

NeedsCompilation no

RoxygenNote 7.3.1

Repository https://boehringer-ingelheim.r-universe.dev

RemoteUrl https://github.com/boehringer-ingelheim/tipmap

RemoteRef HEAD

RemoteSha 99e6c7c65d6f293cbef5572d8c949815d58f4ddf

Contents

`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()] and [create_tipmap_data()].

Examples

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

`create_posterior_data` *Quantiles of 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 or bounds of highest posterior density intervals, generated with varying weights on the informative component of the MAP prior.

Usage

```
create_posterior_data(
  map_prior,
  new_trial_data,
  sigma,
  null_effect = 0,
  interval_type = "equal-tailed",
  n_samples = 10000
)
```

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.
`interval_type`	The type of credible interval (character of length 1), either ‘equal-tailed‘ (default) or ‘hpdi‘, the highest posterior density interval.
`n_samples`	Number of samples to compute highest posterior density intervals (hence, only applicable if the ‘interval_type‘ is ‘hpdi‘).

Details

Highest posterior density intervals are based on ‘coda::HPDinterval()‘ and are an experimental feature.

Value

A data frame containing the default quantiles of posterior mixture distributions or bounds of highest posterior density intervals.

References

Best, N., Price, R. G., Pouliquen, I. J., & Keene, O. N. (2021). Assessing efficacy in important subgroups in confirmatory trials: An example using Bayesian dynamic borrowing. Pharm Stat, 20(3), 551–562. https://doi.org/10.1002/pst.2093

See Also

[create_new_trial_data()] and [create_prior_data()]

Examples

```
# create vector containing data on new trial
new_trial_data <- create_new_trial_data(
  n_total = 30,
  est = 1.27,
  se = 0.95
)

# read MAP prior created by RBesT
map_prior <- load_tipmap_data("tipmapPrior.rds")

# create posterior data - with interval_type = equal_tailed
# (the default for tipping point plots)
posterior_data1 <- create_posterior_data(
  map_prior = map_prior,
  new_trial_data = new_trial_data,
  sigma = 12,
  interval_type = "equal-tailed"
)


# create posterior data - with interval_type = hpdi
posterior_data2 <- create_posterior_data(
  map_prior = map_prior,
  new_trial_data = new_trial_data,
  sigma = 12,
  interval_type = "hpdi",
  n_samples = 1e4
)
```

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()] and [create_posterior_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()] and [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

```
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

```
rweights <- draw_beta_mixture_nsamples(
  n = 50,
  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)
  ),
  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, 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, 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)
  )
)
print(beta_fits)
```

get_cum_probs_1exp *Get cumulative probabilities from distribution of chips of one expert*

Description

Get cumulative probabilities from distribution of chips of one expert

Usage

```
get_cum_probs_1exp(chips)
```

Arguments

chips	Vector of integers, representing the distribution of chips assigned by one expert, as elicited through the roulette method. Each element of the vector represents one bin in the grid.

Value

A numeric vector with the cumulative distribution of chips.

See Also

[get_model_input_1exp()] and [fit_beta_1exp()].

Examples

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

---

get_model_input_1exp *Transform cumulative probabilities to fit beta distributions*

Description

Transform cumulative probabilities to fit beta distributions

Usage

```
get_model_input_1exp(cum_probs, w = NULL)
```

Arguments

cum_probs	Numeric vector, containing cumulative probabilities of weights for one expert, as elicited through the roulette method. Each element of the vector represents one bin in the grid.
w	Numeric vector, upper interval limit of bin (defaults to `1:length(cum_probs) / length(cum_probs)`).

Value

Dataframe to be used as input to fit beta distributions by [fit_beta_1exp()].

See Also

[get_cum_probs_1exp()] and [fit_beta_1exp()].

Examples

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

```
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_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

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

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

```
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

```
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

```
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)
```

---

load_tipmap_data *Load exemplary datasets*

Description

Loads one of three exemplary datasets in the package.

Usage

```
load_tipmap_data(file)
```

Arguments

file	The dataset to be loaded.

Value

A pre-saved dataset.

Examples

```
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

```
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

```
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_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)
```

---

oc_coverage *Assessing coverage*

Description

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

Usage

```
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.
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

```
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)
```

oc_pos *Assessing probability of success*

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

```
oc_pos(
  m,
  se,
  probs,
  weights = seq(0, 1, by = 0.01),
  map_prior,
  sigma,
  null_effect = 0,
  direction_pos = T,
  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).
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(
  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

```
tipmap_darkblue
```

Format

An object of class `character` of length 1.

tipmap_lightred *Custom light red*

Description

Custom light red

Usage

```
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 robust 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

```
tipmap_data <- load_tipmap_data("tipdat.rds")
tipmap_plot(tipmap_data)
```

Index