Package 'mtrank'

June 4, 2025

Title Ranking using Probabilistic Models and Treatment Choice Criteria

Version 0.2-0

Date 2025-06-03

Depends R (>= 4.0.0), meta (>= 8.1-0), netmeta (>= 3.2-0)

Imports PlackettLuce, dplyr, magrittr, ggplot2

Suggests rmarkdown, knitr

Maintainer Theodoros Evrenoglou <theodoros.evrenoglou@uniklinik-freiburg.de>

URL https://github.com/TEvrenoglou/mtrank

Description Estimation of treatment hierarchies in network meta-analysis using a novel frequentist approach based on treatment choice criteria (TCC) and probabilistic ranking models, as described by Evrenoglou et al. (2024) <DOI:10.48550/arXiv.2406.10612>. The TCC are defined using a rule based on the smallest worthwhile difference (SWD). Using the defined TCC, the NMA estimates (i.e., treatment effects and standard errors) are first transformed into treatment preferences, indicating either a treatment preference (e.g., treatment A > treatment B) or a tie (treatment A = treatment B). These treatment preferences are then synthesized using a probabilistic ranking model, which estimates the latent ability parameter of each treatment and produces the final treatment hierarchy. This parameter represents each treatments ability to outperform all the other competing treatments in the network. Here the terms ability to outperform indicates the propensity of each treatment to yield clinically important and beneficial effects when compared to all the other treatments in the network. Consequently, larger ability estimates indicate higher positions in the ranking list.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Theodoros Evrenoglou [aut, cre] (ORCID: <https://orcid.org/0000-0003-3336-8058>), Guido Schwarzer [aut] (ORCID: <https://orcid.org/0000-0001-6214-9087>)

Repository CRAN

Date/Publication 2025-06-04 08:00:09 UTC

Contents

`mtrank-package` *mtrank: Brief overview*

Description

R package **mtrank** enables the estimation of treatment hierarchies in network meta-analysis using a novel frequentist approach based on treatment choice criteria (TCC) and probabilistic ranking models, as described by Evrenoglou et al. (2024). The TCC are defined using a rule based on the smallest worthwhile difference (SWD). Using the defined TCC, the NMA estimates (i.e., treatment effects and standard errors) are first transformed into treatment preferences, indicating either a treatment preference (e.g., treatment A > treatment B) or a tie (treatment A = treatment B). These treatment preferences are then synthesized using a probabilistic ranking model, which estimates the latent ability parameter of each treatment and produces the final treatment hierarchy. This parameter represents each treatments ability to outperform all the other competing treatments in the network. Here the terms "ability to outperform" indicates the propensity of each treatment to yield clinically important and beneficial effects when compared to all the other treatments in the network. Consequently, larger ability estimates indicate higher positions in the ranking list.

Details

The R package **mtrank** provides the following functions:

- Function `tcc` defines the TCC and produces a treatment preference format based on network meta-analysis estimates.
- Function `mtrank` synthesizes the output of the `tcc` function and estimates the final treatment ability.
- Forest plots are created either for the results of the TCC (`forest.tcc`) or the final ability estimates (`forest.mtrank`).
- Function `fitted.mtrank` uses the ability estimates obtained from `mtrank` to calculate pairwise probabilities that any treatment 'A' can be better, equal, or worse than any other treatment 'B' in the network.

- The function linegraph visualizes the output of mtrank across different SWD values. It serves as a sensitivity analysis to the initial choice of SWD.

Type help(package = "mtrank") for a listing of R functions available in **mtrank**.

Type citation("mtrank") on how to cite **mtrank** in publications.

To report problems and bugs, please send an email to Theodoros Evrenoglou <theodoros.evrenoglou@uniklinik-freiburg.de>.

The development version of **mtrank** is available on GitHub https://github.com/TEvrenoglou/mtrank.

Author(s)

Theodoros Evrenoglou <theodoros.evrenoglou@uniklinik-freiburg.de>, Guido Schwarzer <guido.schwarzer@uniklinik-freiburg.de>

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria, https://arxiv.org/abs/2406.10612

See Also

Useful links:

- https://github.com/TEvrenoglou/mtrank

---

antidepressants *Network meta-analysis for major depressive disorder*

Description

Network meta-analysis comparing antidepressants in patients with major depressive disorder.

Format

A data frame with the following columns:

studyid	study id
drug_name	antidepressant name
ntotal	number of randomized patients in treatment arm
responders	number of responders

Source

Cipriani A, Furukawa T, Salanti G, Chaimani A, et al. (2018): Comparative efficacy and acceptability of 21 antidepressant drugs for the acute treatment of adults with major depressive disorder: a systematic review and network meta-analysis *Lancet*, **391**, 1357–66

See Also

mtrank, tcc

Examples

```
data(antidepressants)
head(antidepressants)
#
# Examples:
# example(tcc)
# example(mtrank)
# example(fitted.mtrank)
```

---

diabetes *Network meta-analysis studying the incidence of diabetes*

Description

Network meta-analysis comparing six antihypertensive drugs against the incidence of diabetes.

Format

A data frame with the following columns:

study	study label
id	study id
t	treatment label
r	number of events
n	group sample size
rob	risk of bias assessment

Source

Elliott WJ, Meyer PM (2007): Incident diabetes in clinical trials of antihypertensive drugs: A network meta-analysis *Lancet*, **369**, 201–7, doi:10.1016/S01406736(07)601081

See Also

mtrank, tcc

Examples

```
data(diabetes)
head(diabetes)
#

pw <- pairwise(studlab = study, treat = t,
  n = n, event = r, data = diabetes, sm = "OR")
```

```
#
net <- netmeta(pw, reference.group = "PLA")
#
ranks <- tcc(net, swd = 1.20, small.values = "desirable")
#
forest(ranks)
forest(ranks, reference.group = "ARB", baseline.reference = FALSE)
```

`fitted.mtrank` *Calculate pairwise fitted probabilities for* `mtrank` *object.*

Description

This function uses the estimates of ability and tie prevalence parameters from a `mtrank` object and calculates fitted pairwise probabilities about the preference or the tie between two treatments based on equations (7) and (8) in Evrenoglou et al. (2024).

Usage

```
## S3 method for class 'mtrank'
fitted(object, treat1, treat2, type, ...)

## S3 method for class 'fitted.mtrank'
print(x, type = attr(x, "type"), digits = 4, ...)
```

Arguments

`object`	An object of class `mtrank`.
`treat1`	The first treatment considered in the treatment comparison.
`treat2`	The second treatment considered in the treatment comparison.
`type`	A character vector specifying the probability of interest. Either "better", "tie", "worse", or "all" (can be abbreviated).
`...`	Additional arguments (passed on to `prmatrix`).
`x`	An object of class `fitted.mtrank`.
`digits`	Minimal number of significant digits for proportions, see `print.default`.

Details

Pairwise fitted probabilities between any two treatments in the network can be calculated using the ability estimates obtained from `mtrank` and equations (7) and (8) in Evrenoglou et al. (2024). The fitted probabilities are calculated in the direction `treat1` vs `treat2`. The available probability types are

- "better": probability that `treat1` is better than `treat2`,
- "tie": probability that `treat1` is equal to `treat2`,

- "worse": probability that `treat1` is worse than `treat2`,
- "all": all three probabilities.

Please note that all the arguments of this function are mandatory.

Value

The probability (or probabilities) of interest for the comparison `treat1` vs `treat2` based on the argument `type`.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria, https://arxiv.org/abs/2406.10612

Examples

```
data(antidepressants)
#
pw1 <- pairwise(studlab = studyid, treat = drug_name,
  n = ntotal, event = responders,
  data = antidepressants, sm = "OR")
# Use subset to reduce runtime
pw0 <- subset(pw1, studyid < 60)
#
net0 <- netmeta(pw0, reference.group = "tra")
#
ranks0 <- tcc(net0, swd = 1.20, small.values = "undesirable")
#
fit0 <- mtrank(ranks0)
#
fitted(fit0, type = c("better", "worse"),
  treat1 = "bupropion", treat2 = "escitalopram")
#
fitted(fit0, type = c("better", "worse"),
  treat1 = "escitalopram", treat2 = "bupropion")
#
fitted(fit0, type = "all",
  treat1 = c("bupropion", "escitalopram"),
  treat2 = c("escitalopram", "bupropion"))

## Not run:
# Run analysis with full data set
net1 <- netmeta(pw1, reference.group = "tra")
#
ranks1 <- tcc(net1, swd = 1.20, small.values = "undesirable")
#
fit1 <- mtrank(ranks1)
#
fitted(fit1, type = c("better", "worse"),
  treat1 = "bupropion", treat2 = "escitalopram")
```

```
#
fitted(fit1, type = c("better", "worse"),
  treat1 = "escitalopram", treat2 = "bupropion")
#
fitted(fit1, type = "all",
  treat1 = c("bupropion", "escitalopram"),
  treat2 = c("escitalopram", "bupropion"))

## End(Not run)
```

`forest.mtrank` *Forest plot of ability estimates produced with* `mtrank`

Description

This function produces a forest plot that visualizes the ability estimates calculated with `mtrank`.

Usage

```
## S3 method for class 'mtrank'
forest(
  x,
  sorting = "ability",
  backtransf = FALSE,
  xlab = "",
  leftcols = "studlab",
  leftlabs = "Treatment",
  rightcols = c("effect", "ci"),
  rightlabs = c(paste0(if (!backtransf) "log-", "Abilities"), NA),
  label.left = "Favors average treatment",
  label.right = "Favors treatment",
  header.line = TRUE,
  ...
)
```

Arguments

`x`	An object of class `mtrank`.
`sorting`	An argument specifying the criterion to sort the ability estimates in the forest plot (see Details).
`backtransf`	A logical argument specifying whether to show log-ability estimates (`FALSE`, default) or ability estimates on the natural scale (`TRUE`).
`xlab`	A label for the x-axis.
`leftcols`	A character vector specifying columns to be printed on the left side of the forest plot (see `forest.meta`).

leftlabs	A character vector specifying labels for columns on left side of the forest plot.
rightcols	A character vector specifying columns to be printed on the right side of the forest plot (see forest.meta).
rightlabs	A character vector specifying labels for columns on right side of the forest plot.
label.left	Graph label on left side of null effect.
label.right	Graph label on right side of null effect.
header.line	A logical value indicating whether to print a header line or a character string ("both", "below", "").
...	Additional arguments (passed on to forest.meta).

Details

The function produces a forest plot and visualizes the ability estimates obtained from mtrank. The order of the estimates in the forest plot (argument sorting) can be one of the following:

- "ability": sort by descending ability estimates (default),
- "se": sort by descending precision, i.e., increasing standard errors,
- "none": use order from data set.

Value

A forest plot is plotted in the active graphics device.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria, https://arxiv.org/abs/2406.10612

Examples

```
# Examples: example(mtrank)
```

---

forest.tcc *Forest plot showing the treatment preference format of the NMA estimates according to treatment choice criterion.*

Description

This function produces a forest plot and visualizes the treatment preference format of the NMA estimates as defined from the treatment choice criterion in tcc.

Usage

```
## S3 method for class 'tcc'
forest(
  x,
  reference.group = x$reference.group,
  baseline.reference = x$baseline.reference,
  backtransf = FALSE,
  leftcols = "studlab",
  leftlabs,
  rightcols = c("effect", "ci"),
  col.winner = "red",
  col.tie = "black",
  lty.equi = gs("lty.cid"),
  col.equi = gs("col.cid"),
  fill.equi = "lightblue",
  fill.swd.below.null = "transparent",
  fill.swd.above.null = "transparent",
  smlab,
  header.line = TRUE,
  ...
)
```

Arguments

x	An object of class tcc.
reference.group	Reference treatment(s). By default, the graph plots the NMA estimates of all treatments versus the common reference treatment used in the netmeta object.
baseline.reference	A logical indicating whether results should be expressed as comparisons of other treatments versus the reference treatment (default) or vice versa.
backtransf	A logical indicating whether results should be back transformed. If backtransf = TRUE (default), results for sm = "OR" are printed as odds ratios rather than log odds ratios, for example.
leftcols	A character vector specifying columns to be printed on the left side of the forest plot (see forest.meta).
leftlabs	A character vector specifying labels for columns on left side of the forest plot.
rightcols	A character vector specifying columns to be printed on the right side of the forest plot (see forest.meta).
col.winner	Colour to highlight results for TCC winner.
col.tie	Colour to highlight results for TCC ties.
lty.equi	Line type (limits of equivalence).
col.equi	Line colour (limits of equivalence).
fill.equi	Colour(s) for area between limits of equivalence.

`fill.swd.below.null`
: Colour of area below lower SWD limit.

`fill.swd.above.null`
: Colour of area above upper SWD limit.

`smlab`
: A label for the summary measure (printed at top of figure).

`header.line`
: A logical value indicating whether to print a header line or a character string ("both", "below", "").

`...`
: Additional arguments (passed on to `forest.meta`).

Details

This function produces forest plots for the NMA treatment effect estimates. The color indicates whether treatment effects show a preference (red color) or tie (black color). Additionally, the respective range of equivalence defined at the function `tcc` is visualized for the forest plot.

The argument `reference.group` is optional. By default, the graph plots the NMA estimates of all treatments versus the common reference treatment used in the `netmeta` object.

Value

A forest plot is plotted in the active graphics device.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria, https://arxiv.org/abs/2406.10612

Examples

```
data("antidepressants")
#
pw1 <- pairwise(studlab = studyid, treat = drug_name,
  n = ntotal, event = responders,
  data = antidepressants, sm = "OR")
# Use subset to reduce runtime
pw0 <- subset(pw1, studyid < 60)
#
net0 <- netmeta(pw0, reference.group = "tra")

ranks0 <- tcc(net0, swd = 1.20, small.values = "undesirable")

# Comparison other drugs vs trazodone
forest(ranks0,
  label.left = "Favours trazodone",
  label.right = "Favours other drug")

# Comparison escitalopram vs other drugs
forest(ranks0, reference.group = "esc", baseline = FALSE,
  label.left = "Favours other drug",
  label.right = "Favours escitalopram")
```

```
## Not run:
# Store a PDF file in the current working directory showing all results
# (this is the default, i.e., if argument 'reference.group' is missing)
forest(ranks0, baseline = FALSE, reference.group = trts,
  file = "forest_tcc_antidepressants.pdf")

# Run analysis with full data set
net1 <- netmeta(pw1, reference.group = "tra")

ranks1 <- tcc(net1, swd = 1.20, small.values = "undesirable")

# Comparison other drugs vs trazodone
forest(ranks1,
  label.left = "Favours trazodone",
  label.right = "Favours other drug")

# Comparison escitalopram vs other drugs
forest(ranks1, reference.group = "esc", baseline = FALSE,
  label.left = "Favours other drug",
  label.right = "Favours escitalopram")

## End(Not run)
```

`linegraph` *Line graph showing the results of* `mtrank` *across different smallest worthwhile difference (SWD) values*

Description

This function produces a line graph that visualizes the results of `mtrank` in terms of either abilities or probabilities across different smallest worthwhile difference (SWD) values.

Usage

```
linegraph(
  x,
  swd,
  swd.ref,
  small.values = x$small.values,
  type = "probability",
  k = length(x$trts),
  backtransf = FALSE,
  linewidth = 1.1,
  point.size = 2,
  ...
)
```

Arguments

x	An object of class mtrank.
swd	A numeric vector of SWD values to be used for the sensitivity analysis.
swd.ref	A numeric SWD value to be used as the reference for sorting treatments in the final graph. This value must be included in swd.
small.values	A character string specifying whether small treatment effects indicate a beneficial ("desirable") or harmful ("undesirable") effect; can be abbreviated.
type	The metric to be used for plotting the results of the sensitivity analysis. Two options are available: the default is "probability", which plots results in terms of normalized abilities; the alternative is "ability", which plots results in terms of ability estimates. Both options can be abbreviated.
k	A numeric value indicating the number of treatments to be plotted. By default, all available treatments are shown. For large networks, it is advisable to limit the number of treatments to improve readability. If specified, the first k treatments based on the hierarchy at swd.ref will be plotted.
backtransf	A logical value indicating whether to display log-ability estimates (FALSE, default) or back-transformed ability estimates on the natural scale (TRUE). This argument is ignored if type = "probability".
linewidth	A numeric value specifying the width of the lines (default: 1.1).
point.size	A numeric value specifying the size of the points (default: 2).
...	Additional arguments passed to mtrank.

Details

This function creates a line graph to visualize probability or ability estimates obtained from mtrank across different SWD values. The order of treatments in the graph is based on their hierarchy at the reference SWD value (swd.ref).

Value

A ggplot object.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Ruecker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria, https://arxiv.org/abs/2406.10612

Examples

```
data("antidepressants")
#
pw <- pairwise(studlab = studyid, treat = drug_name,
  n = ntotal, event = responders,
  data = antidepressants, sm = "OR")
# Use subset to reduce runtime
pw <- subset(pw, studyid < 60)
```

```
#
net <- netmeta(pw, reference.group = "tra")
#
ranks <- tcc(net, swd = 1.20, small.values = "undesirable")
#
fit <- mtrank(ranks)
#
# Perform a sensitivity analysis across different SWD values assuming that
# 1.20 is the reference value
swd.vec <- seq(1.10, 1.50, by = 0.10)
swd.ref <- 1.20
# plot all the treatments in the network
linegraph(fit, swd = swd.vec, swd.ref = swd.ref)

# plot only the first three treatments in the order appearing at the
# 'swd.ref' value
linegraph(fit, swd = swd.vec, swd.ref = swd.ref, k = 3)
# plot in terms of ability estimates
linegraph(fit, swd = swd.vec, swd.ref = swd.ref, type = "ability")
```

`mtrank` *Estimate the treatment hierarchy in network meta-analysis using a probabilistic ranking model*

Description

This function fits the Davidson-Bradley-Terry ranking model and produces a treatment hierarchy based on the method described by Evrenoglou et al. (2024) for network meta-analysis.

Usage

```
mtrank(x, level = x$level, ...)

## S3 method for class 'mtrank'
print(
  x,
  sorting = "ability",
  backtransf = FALSE,
  digits = gs("digits"),
  digits.prop = gs("digits.prop"),
  ...
)
```

Arguments

`x`	An object of class `tcc` or `mtrank` (print function).
`level`	The level used to calculate confidence intervals for ability estimates.

...	Additional arguments (passed on to PlackettLuce or to prmatrix).
sorting	An argument specifying the criterion to sort the ability estimates in the printout (see Details).
backtransf	A logical argument specifying whether to show log-ability estimates (FALSE, default) or on the natural scale (TRUE).
digits	Minimal number of significant digits for ability estimates, see print.default.
digits.prop	Minimal number of significant digits for proportions, see print.default.

Details

This function fits a Davidson-Bradley-Terry model to the treatment preferences tcc function. It estimates the ability of each treatment to outperform the other treatments in the network, along with the respective standard errors, using a maximum likelihood approach. The term 'ability to outperform' refers to a latent characteristic that indicates the propensity of each treatment in the network to yield clinically relevant and beneficial treatment effects, in the context of the defined treatment choice criterion, when compared to the rest of the treatments. Consequently, treatments with larger ability estimates are ranked more prominently in the treatment hierarchy.

To retain identifiability, the maximization of the log-likelihood takes place subject to the constrain that the ability estimates sum to 1. Then, the maximum likelihood estimates (MLEs) are calculated iteratively. Note that the final estimates of the ability parameters are not necessarily needed to sum to 1 as after the first iteration of the algorithm the ability estimates are not normalized. However, by normalizing the final ability estimates to sum up to 1 these can be interpreted as "the probability that each treatment is having the highest ability".

Finally, a parameter "v" controlling the prevalence of ties in the network is also estimated. Although the estimated values of this parameter do not have a direct interpretation they are useful for estimating the fitted pairwise probabilities (see fitted.mtrank).

Value

- A data frame containing the resulting log-ability estimates, their standard errors and their confidence intervals.
- The estimate of the tie prevalence parameter v, on the log-scale.
- The normalized ability estimates for each treatment.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria, https://arxiv.org/abs/2406.10612

Examples

```
data("antidepressants")
#
pw <- pairwise(studlab = studyid, treat = drug_name,
  n = ntotal, event = responders,
  data = antidepressants, sm = "OR")
```

```
# Use subset to reduce runtime
pw <- subset(pw, studyid < 60)
#
net <- netmeta(pw, reference.group = "tra")

ranks <- tcc(net, swd = 1.20, small.values = "undesirable")
#
fit <- mtrank(ranks)

# Print log-ability estimates
fit
#
# Print ability estimates
print(fit, backtransf = TRUE)

# Visualize results
forest(fit)
```

`pp2long` *Auxiliary function to transform data from paired-preference to long-arm format*

Description

Auxiliary function to transform data from paired-preference to long-arm format

Usage

```
pp2long(x)
```

Arguments

`x`	An object of class "ppdata" (part of `tcc` object).

Value

Data set in long-arm format that can be used as input to `rankings`.

Author(s)

Guido Schwarzer <guido.schwarzer@uniklinik-freiburg.de>

See Also

`tcc`, `rankings`

Examples

```
data(diabetes)
#
pw <- pairwise(studlab = study, treat = t,
  n = n, event = r, data = diabetes, sm = "OR")
# Use subset to reduce runtime
pw1 <- subset(pw, id >= 6 & id <= 10)
net1 <- netmeta(pw1, reference.group = "PLA")
#
ranks1 <- tcc(net1, swd = 1.20, small.values = "desirable")
#
pdat1 <- ranks1$ppdata
#
ldat1 <- pp2long(pdat1)
head(ldat1)


net <- netmeta(pw, reference.group = "PLA")
#
ranks <- tcc(net, swd = 1.20, small.values = "desirable")
#
pdat <- ranks$ppdata
#
ldat <- pp2long(pdat)
head(ldat)

library("PlackettLuce")
preferences <- rankings(ldat, id = "id", item = "treat", rank = "rank")
#
fit <- PlackettLuce(preferences)
#
coef(summary(fit, ref = ranks$reference.group))[, 1]
# Results stored in mtrank()
mtrank(ranks)$estimates$log_ability
```

`tcc` *Apply a treatment-choice criterion (TCC) to get treatment preferences based on network meta-analysis estimates.*

Description

This function uses a treatment choice criterion defined by the user and transforms the network meta-analysis estimates into a preference format that indicates either a treatment preference or a tie. In this setting, a treatment preference implies that the respective NMA estimate represents a clinically important result (i.e. that fulfills the TCC) while a tie indicates that the respective NMA estimate lacks enough evidence to represent a treatment preference. The resulting preference format is then used as input to `mtrank`.

Usage

```
tcc(
  x,
  pooled = if (x$random) "random" else "common",
  swd = NULL,
  swd.below.null = NULL,
  swd.above.null = NULL,
  small.values = x$small.values,
  relax = TRUE,
  level = x$level.ma
)

## S3 method for class 'tcc'
print(x, ...)
```

Arguments

x	A netmeta object.
pooled	A character string indicating whether results for the common ("common") or random effects model ("random") should be used. Can be abbreviated. If not specified the results from the random effects model will be used by default.
swd	A numeric value specifying the smallest worthwhile difference value (SWD); see Details.
swd.below.null	A numeric value specifying the SWD below the null effect (see Details).
swd.above.null	A numeric value specifying the SWD above the null effect (see Details).
small.values	A character string specifying whether small treatment effects indicate a beneficial ("desirable") or harmful ("undesirable") effect.
relax	A logical optional argument. If TRUE (default), the treatment choice criterion is based solely on the SWD bounds, emphasizing only the clinical importance of the results. If set to FALSE, the criterion incorporates both statistical significance and clinical importance. We recommend using the default setting (see Details).
level	The level used to calculate confidence intervals for log-abilities.
...	Additional arguments (ignored).

Details

R function mtrank expects data in a **preference** format, where a treatment preference or tie is indicated for each network meta-analysis (NMA) estimate. For example, for the comparison between treatments A and B the potential outcomes are:

- $A > B$
- $A < B$
- $A = B$

The transformation takes place based on the NMA estimates and the treatment choice criterion which has the form of a decision rule.

This function implements treatment choice criteria based on the range of equivalence (ROE) which are specified by

- argument `swd`. Then the limits of the ROE will be defined based on the values (i) `swd`, `1 / swd` for ratio measures and (ii) `swd` and `-swd` for difference measures.
- arguments `swd.below.null` and `swd.above.null`. These arguments allow the users to define their own limits of the ROE, given the restriction that the lower limit will always be smaller than the upper limit.

Note that when the argument `swd` is specified, the arguments `swd.below.null` and `swd.above.null` are ignored. Either only the `swd` or both of the `swd.below.null` and `swd.above.null` must be specified for the proper definition of the ROE.

After setting the ROE, each NMA treatment effect will be categorised as a treatment preference or a tie. The argument `relax` controls the amount of conservatism of the treatment choice criterion. If set to `FALSE`, a TCC will be built requiring both clinical importance as statistical significance of the results. If set to `TRUE` (default), the criterion uses only the ROE bounds and therefore the NMA treatment effects need to be only clinically important to indicate a treatment preference.

Value

NMA estimates in a preference format.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria, https://arxiv.org/abs/2406.10612

Examples

```
data("antidepressants")
#
pw1 <- pairwise(studlab = studyid, treat = drug_name,
  n = ntotal, event = responders,
  data = antidepressants, sm = "OR")
# Use subset to reduce runtime
pw0 <- subset(pw1, studyid < 60)
#
net0 <- netmeta(pw0, reference.group = "tra")

ranks0 <- tcc(net0, swd = 1.20, small.values = "undesirable")

# Comparison other drugs vs trazodone
forest(ranks0,
  label.left = "Favours trazodone",
  label.right = "Favours other drug")

# Comparison escitalopram vs other drugs
```

```
forest(ranks0, reference.group = "esc", baseline = FALSE,
  label.left = "Favours other drug",
  label.right = "Favours escitalopram")

## Not run:
# Store a PDF file in the current working directory showing all results
# (this is the default, i.e., if argument 'reference.group' is missing)
forest(ranks0, baseline = FALSE, reference.group = trts,
  file = "forest_tcc_antidepressants.pdf")

# Run analysis with full data set
net1 <- netmeta(pw1, reference.group = "tra")

ranks1 <- tcc(net1, swd = 1.20, small.values = "undesirable")

# Comparison other drugs vs trazodone
forest(ranks1,
  label.left = "Favours trazodone",
  label.right = "Favours other drug")

# Comparison escitalopram vs other drugs
forest(ranks1, reference.group = "esc", baseline = FALSE,
  label.left = "Favours other drug",
  label.right = "Favours escitalopram")

## End(Not run)
```

Index