Package: mvnma (via r-universe)

October 18, 2026

Title Multivariate Network Meta-Analysis using Bayesian Methods

Version 0.3-0

Date 2026-09-18

Depends R (>= 4.0.0), meta (>= 8.5-0), netmeta (>= 3.7-0)

Imports R2jags, coda, dplyr, magrittr, matrixStats, rlist, ggplot2, forcats

Maintainer Theodoros Evrenoglou

<theodoros.evrenoglou@uniklinik-freiburg.de>

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

Description Tools to conduct Bayesian multivariate network meta-analysis providing - the single correlation coefficient model by Efthimiou et al. (2015) <doi:10.1093/biostatistics/kxu030>; - per-outcome consistency checks using the node-splitting method (Dias et al., 2010) <doi:10.1002/sim.3767>; - per-outcome treatment hierarchies using the surface under the cumulative ranking curve (SUCRA), the probability of best value, or median (or mean) ranks (Salanti et al., 2011) <doi:10.1016/j.jclinepi.2010.03.016>; - across-outcomes benefit-risk assessment using the ViševKriterijumska Optimizacija I Kompromisno Rešenje (VIKOR) method (Opricovic & Tzeng, 2004) <doi:10.1016/S0377-2217(03)00020-1>; - convergence checks using trace plots, density plots, or the R-hat statistic; - forest plots of treatment estimates and consistency checks, scatter plots of per-outcome rankings, Hasse diagrams (Carlsen & Bruggemann, 2014) <doi:10.1002/cem.2569> to visualize the partial order of the treatments across all outcomes.

License GPL (>= 2)

Encoding UTF-8

NeedsCompilation no

Config/roxygen2/version 8.1.0

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 https://cran.r-universe.dev

Date/Publication 2026-09-18 16:34:46 UTC

RemoteUrl https://github.com/cran/mvnma

RemoteRef HEAD

RemoteSha 70c6b757b5d01efaf44237eeeed1dd8c8d007445

Contents

mvnma-package 2
as.mcmc.mvnma 5
forest.mvnma 6
forest.netsplit.mvnma 8
hasse.mvrank 11
heatplot.mvrank 12
linechart 14
mvnma 15
mvrank 20
netsplit.mvnma 22
plot.mvrank 26
spiechart.mvrank 27
vikor.mvrank 28

Index **31**

`mvnma-package` *mvnma: Brief overview of methods*

Description

R package **mvnma** provides R functions for Bayesian multivariate network meta-analysis (mvNMA). The mvNMA model supported by this package refers to the single correlation coefficient model, interpreted as an amalgam of within- and across-outcome correlations (Efthimiou et al., 2015), which is a generalisation of Riley et al. (2008). In this way, the model does not depend on the extraction of within-study outcome correlations, which are seldom reported at the study level. Note that mvnma can be used to conduct a joint analysis across any number of outcomes and study arms.

The model assumption of consistency is checked by performing the node-splitting method (Dias et al., 2010; König et al., 2013). Within each outcome, and for each pairwise comparison for which both a direct and an indirect estimate can be calculated, the method statistically evaluates whether their difference is equal to zero.

The treatment effect estimates and confidence intervals can be summarised both in terms of per-outcome treatment hierarchies and in terms of an across-outcomes benefit-risk assessment. The former is achieved using ranking methods such as the surface under the cumulative ranking curve

(SUCRA) (Salanti et al., 2011), the probability of being the best treatment, and median (or mean) ranks, each accompanied by a credible interval.

A benefit-risk assessment is possible through the VišeKriterijumska Optimizacija I Kompromisno Rešenje (VIKOR) method (Opricovic & Tzeng, 2004; Opricovic, 2011). This approach, originally proposed in the field of multi-criteria decision analysis, uses a deterministic algorithm to derive an amalgamated treatment hierarchy across outcomes and to explicitly identify the set of treatments that offer the best compromise between benefits and harms across all outcomes.

Since the output of the method relies on Markov Chain Monte Carlo (MCMC) sampling, convergence can be assessed using a series of diagnostics, including trace plots, density plots, and the R-hat statistic. Finally, this package offers the option to visualise the results of the mvNMA model through forest plots, which display the treatment effect estimates and consistency checks; scatter plots, which show the per-outcome rankings for any pair of outcomes; and Hasse diagrams (Carlsen & Bruggemann, 2014), which visualise the partial order of treatments across all outcomes (Rücker & Schwarzer, 2017), as well as alternative methods for deriving a between-outcomes hierarchy, such as the spie chart method (Daly et al., 2020).

Details

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

- Function `mvnma` to perform a Bayesian multivariate network meta-analysis.
- Function `netsplit.mvnma` to perform local checks for inconsistency using the node-splitting method.
- Function `mvrank` to get outcome-specific treatment rankings.
- Function `vikor.mvrank` to rank treatments across all outcomes using the VIKOR multi-criteria decision analysis method. Additionally, the function evaluates the concrete conditions defined by the VIKOR method and identifies the set of treatments that offer the best compromise between benefits and harms across all outcomes.
- Function `forest.mvnma` to visualize the results of the mvNMA model in terms of treatment effect estimates.
- Function `forest.netsplit.mvnma` to visualize the results of the node-splitting consistency checks.
- Function `plot.mvrank` to visualize per outcome ranking results for any pair of outcomes.
- Function `hasse.mvrank` to visualize the partial order of the treatment across all outcomes.
- Function `heatplot.mvrank` to visualize in a heatplot the results in terms of outcome specific rankings.
- Function `linechart` to visualize the results of the three metrics calculated by the VIKOR method.
- Function `as.mcmc.mvnma` an auxiliary function to extract an MCMC object. This makes any **mvnnma** object compatible with the convergence checks performed by the R package **coda**.
- Function `spiechart` to calculate a between outcomes hierarchy using the spie charts method.

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

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

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

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

Author(s)

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

References

Carlsen L, Bruggemann R (2014): Partial order methodology: a valuable tool in chemometrics. *Journal of Chemometrics*, **28**, 226–34

Daly CH, Mbuagbaw L, Thabane L, Straus SE, Hamid JS (2020): Spie charts for quantifying treatment effectiveness and safety in multiple outcome network meta-analysis: a proof-of-concept study. *BMC Med Res Methodol*, **20**, 266

DuMouchel WH, Harris JE (1983): Bayes methods for combining the results of cancer studies in humans and other species. *Journal of the American Statistical Association*, **78**, 293–308

Efthimiou O, Mavridis D, Riley RD, Cipriani A, Salanti G (2015): Joint synthesis of multiple correlated outcomes in networks of interventions. *Biostatistics*, **16**, 84–97

Opricovic S, Tzeng GH (2004): Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European Journal of Operational Research*, **156**, 445–55

Opricovic S (2011): Fuzzy VIKOR with an application to water resources planning. *Expert Systems with Applications*, **38**, 12983–90

Riley RD, Thompson JR, Abrams KR (2008): An alternative model for bivariate random-effects meta-analysis when the within-study correlations are unknown. *Biostatistics*, **9**, 172–86

Rücker G, Schwarzer G (2017): Resolve conflicting rankings of outcomes in network meta-analysis: Partial ordering of treatments. *Research Synthesis Methods*, **8**, 526–36

Salanti G, Ades AE, Ioannidis JP (2011): Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *Journal of Clinical Epidemiology*, **64**, 163–71

Dias S, Welton NJ, Caldwell DM, Ades AE (2010): Checking consistency in mixed treatment comparison meta-analysis. *Statistics in Medicine*, **29**, 932–44

König J, Krahn U, Binder H (2013): Visualizing the flow of evidence in network meta-analysis and characterizing mixed treatment comparisons *Statistics in Medicine*, **32**, 5414–29

See Also

Useful links:

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

as.mcmc.mvnma *Auxiliary function to extract the Markov Chain Monte Carlo object*

Description

Extract the Markov Chain Monte Carlo object from an R object of class mvnma.

Usage

```
## S3 method for class 'mvnma'
as.mcmc(x, drop.reference.group = FALSE, ...)
```

Arguments

x	An object of class mvnma.
drop.reference.group	A logical indicating whether to drop the MCMC samples for the reference group.
...	Additional arguments passed on to as.mcmc.list.

Value

The function returns the Markov Chain Monte Carlo object which is a list of 'mcmc' objects.

Author(s)

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

See Also

mvnma

Examples

```
# Locate file "mvnma_examples.rda" with mvnma() results
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

mcmc12 <- as.mcmc(mvnma12)

library(coda)
oldpar <- par(mfrow = c(3, 3))
# Traceplot
traceplot(mcmc12)
# Density plot
densplot(mcmc12)
# Traceplot and density plot together
plot(mcmc12)

# Do not print the trace plot for the reference group, i.e., placebo
```

```
mcmc12.drop <- as.mcmc(mvnma12, drop.reference.group = TRUE)
par(mfrow = c(2, 4))
# traceplot
traceplot(mcmc12.drop)
# density plot
densplot(mcmc12.drop)
# traceplot and density plot together
plot(mcmc12.drop)

par(oldpar)
```

`forest.mvnma` *Forest plot for multivariate network meta-analysis results*

Description

Draws a forest plot in the active graphics window (using grid graphics system).

Usage

```
## S3 method for class 'mvnma'
forest(
  x,
  backtransf = FALSE,
  separate = n_distinct(attr(x, "sm")) > 1,
  leftcols = "studlab",
  leftlabs,
  rightcols = c("effect", "ci"),
  rightlabs,
  col.study = "black",
  col.square = "black",
  col.square.lines = "black",
  col.subgroup = "black",
  squaresize = 0.7,
  header.line = TRUE,
  text.outcome = "Outcome: ",
  ...
)
```

Arguments

`x`	An object of class `mvnma`.
`backtransf`	A logical indicating whether results should be back transformed in forest plots. If `backtransf = TRUE`, results for `sm = "OR"` are presented as odds ratios rather than log odds ratios, for example.
`separate`	A logical indicating whether separate forest plots should be created for outcomes.

leftcols	A character vector specifying (additional) columns to be printed on the left side of the forest plot or a logical value.
leftlabs	A character vector specifying labels for (additional) columns on left side of the forest plot.
rightcols	A character vector specifying (additional) columns to be printed on the right side of the forest plot or a logical value.
rightlabs	A character vector specifying labels for (additional) columns on right side of the forest plot.
col.study	The colour for treatment effect estimates and confidence interval limits.
col.square	The colour for squares reflecting the treatment effect estimates.
col.square.lines	The colour for the outer lines of squares reflecting the treatment effect estimates.
col.subgroup	The colour to print information on different outcomes.
squaresize	The size of squares reflecting the treatment effect estimates (default: 0.7).
header.line	A logical value indicating whether to print a header line (default: TRUE) or a character string ("both", "below", "").
text.outcome	A text string printed before the name of the outcome.
...	Additional arguments passed on to forest.meta.

Value

A forest plot is shown in the active graphics window.

Examples

```
# Locate file "mvnma_examples.rda" with mvnma() results
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

# Print multivariate network meta-analysis results
mvnma12

# Generate a forest plot with the results
forest(mvnma12)

# Generate a separate forest plot for each outcome
forest(mvnma12, separate = TRUE)
```

forest.netsplit.mvnma *Forest plot for direct and indirect evidence in multivariate network meta-analysis*

Description

Forest plot showing the multivariate network meta-analysis estimate, the direct estimate and the indirect estimate for each treatment comparison that has been node-split. A separate forest plot is produced for each outcome.

Usage

```
## S3 method for class 'netsplit.mvnma'
forest(
  x,
  outcome = NULL,
  show = "both",
  subgroup = "comparison",
  overall = TRUE,
  direct = TRUE,
  indirect = TRUE,
  sortvar = NULL,
  subset = NULL,
  text.overall = "mvNMA estimate",
  text.direct = "Direct estimate",
  text.indirect = "Indirect estimate",
  type.overall,
  type.direct,
  type.indirect,
  col.square = "gray",
  col.square.lines = col.square,
  col.diamond = "gray",
  col.diamond.lines = "black",
  equal.size = TRUE,
  leftcols,
  leftlabs,
  rightcols = c("effect", "ci"),
  rightlabs = NULL,
  digits = gs("digits.forest"),
  digits.prop = max(gs("digits.pval") - 2, 2),
  backtransf = gs("backtransf"),
  lab.NA = "",
  smlab,
  file = NULL,
  ...
)
```

Arguments

x	An object of class netsplit.
outcome	An optional character vector or numeric vector selecting the outcome(s) to plot. By default, a forest plot is produced for every outcome with at least one comparison providing both direct and indirect evidence.
show	A character string indicating which comparisons should be plotted: "both" (comparisons with both direct and indirect evidence), "all", "with.direct" (comparisons with direct evidence), "direct.only", or "indirect.only".
subgroup	A character string indicating which layout should be used in the forest plot: subgroups by treatment comparison ("comparison") or by type of estimate ("estimate").
overall	A logical indicating whether estimates from the multivariate network meta-analysis should be plotted.
direct	A logical indicating whether estimates derived from direct evidence should be plotted.
indirect	A logical indicating whether estimates derived from indirect evidence should be plotted.
sortvar	An optional vector used to sort the treatment comparisons.
subset	An optional logical vector selecting a subset of the treatment comparisons.
text.overall	A character string used in the forest plot to label the multivariate network meta-analysis estimate.
text.direct	A character string used in the forest plot to label the direct estimate.
text.indirect	A character string used in the forest plot to label the indirect estimate.
type.overall	A character string specifying how to plot the multivariate network meta-analysis estimate.
type.direct	A character string specifying how to plot the direct estimate.
type.indirect	A character string specifying how to plot the indirect estimate.
col.square	The colour for squares.
col.square.lines	The colour for the outer lines of squares.
col.diamond	The colour for diamonds.
col.diamond.lines	The colour for the outer lines of diamonds.
equal.size	A logical indicating whether all squares should be of equal size.
leftcols	A character vector specifying (additional) columns to be printed on the left side of the forest plot.
leftlabs	A character vector specifying labels for the columns on the left side of the forest plot.
rightcols	A character vector specifying (additional) columns to be printed on the right side of the forest plot.
rightlabs	A character vector specifying labels for the columns on the right side of the forest plot.

digits	Minimal number of significant digits for treatment estimates and confidence intervals, see print.default.
digits.prop	Minimal number of significant digits for direct evidence proportions, see print.default.
backtransf	A logical indicating whether results should be back transformed in the forest plot. For example, if backtransf = TRUE, results for sm = "OR" are shown as odds ratios rather than log odds ratios.
lab.NA	A character string to label missing values.
smlab	A label for the summary measure. By default, the name of the outcome is used.
file	An optional character string with the name of a file to save the forest plot(s), see forest.meta. If more than one outcome is plotted, the outcome name is appended to the file name, e.g., "forest_Early_Response.pdf".
...	Additional arguments passed on to forest.meta.

Details

A separate forest plot is produced for each selected outcome, since the set of comparisons providing both direct and indirect evidence can differ between outcomes. An outcome is skipped, with a warning, if it holds no such comparison or if argument show selects none of them.

Estimates are stored in the netsplit object on the original scale, so argument backtransf controls the scale used in the forest plot.

If argument file is provided, one file is written per outcome, and the outcome name is appended to the file name when several outcomes are plotted so that the files do not overwrite each other. To collect all plots in a single multi-page file, open a graphics device before calling this function instead of using argument file.

Value

A list with the data sets used to produce the forest plot(s), returned invisibly, with one element per selected outcome. Each element holds the estimates plotted for that outcome, on the original scale. Skipped outcomes are NULL.

See Also

netsplit.mvnma

Examples

```
## Not run:
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

ns <- netsplit(mvnma_all)
forest(ns)

## End(Not run)
```

`hasse.mvrank` *Hasse diagram*

Description

This function generates a Hasse diagram for a partial order of treatment ranks in a multivariate network meta-analysis.

Usage

```
## S3 method for class 'mvrank'
hasse(x, ...)
```

Arguments

`x`	An object of class `mvrank`.
`...`	Additional arguments passed on to `hasse.netposet`.

Details

Generate a Hasse diagram (Carlsen & Bruggemann, 2014) for a partial order of treatment ranks in a network meta-analysis (Rücker & Schwarzer, 2017).

This R function is a wrapper function for `hasse.netposet`.

Value

A Hasse diagram is shown in the active graphics window.

References

Carlsen L, Bruggemann R (2014): Partial order methodology: a valuable tool in chemometrics. *Journal of Chemometrics*, **28**, 226–34

Rücker G, Schwarzer G (2017): Resolve conflicting rankings of outcomes in network meta-analysis: Partial ordering of treatments. *Research Synthesis Methods*, **8**, 526–36

Examples

```
# Locate file "mvnma_examples.rda" with mvnma() results
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

# Print odds ratios for efficacy outcomes
outc <- names(mvnma12)[names(mvnma12) != "cor"]
#
for (i in outc) {
  cat(paste0("\nOutcome: ", i, "\n\n"))
  print(round(exp(mvnma12[[i]]$TE.random), 2))
}
```

```
# Rank treatments using SUCRAs
ranks12 <- mvrank(mvnma12, method = "sucra",
  small.values = c("undes", "undes"))
ranks12

# Get the Hasse diagram for the efficacy outcomes
hasse(ranks12)
```

`heatplot.mvrank` *A heatplot to visualize the ranking in a multivariate network meta-analysis*

Description

This function produces a heatplot displaying the results of function `mvrank`. The graph can be used to visualize the ranking output when the method used to rank the treatments is either the surface under the cumulative ranking curve (SUCRA) or the probability of being best (pbest).

Usage

```
## S3 method for class 'mvrank'
heatplot(
  x,
  sort = NULL,
  col.num = "white",
  num.size = 4.5,
  col.low = "lightblue",
  col.high = "darkblue",
  width.bar = 1.2,
  angle.x = 0,
  angle.y = 0,
  hjust.x = 0.5,
  hjust.y = 1,
  hjust.legend = 0.5,
  size.x = 8,
  size.y = 10,
  legend.position = "right",
  legend.direction = "vertical",
  ...
)
```

Arguments

`x`	An object of class `mvrank`.

sort	An optional argument to define an outcome to be used as a reference when sorting the order of treatments on the x-axis. If NULL, then the first outcome, as it appears in mvrank, is used as the reference. The user can either provide a single number corresponding to the position of the outcome in the mvrank object or a character string with the name of the outcome; can be abbreviated.
col.num	The color of the numbers in the squares (default: "white").
num.size	The size of the numbers in the squares (default: 4.5).
col.low	The color for the low end of the gradient (default: "lightblue").
col.high	The color for the high end of the gradient (default: "darkblue").
width.bar	The width of the bars in the graph (default: 1.2).
angle.x	The angle (in [0, 360]) to rotate the text appearing on the x-axis (default: 0).
angle.y	The angle (in [0, 360]) to rotate the text appearing on the y-axis (default: 0).
hjust.x	Horizontal justification for the text on the x-axis (default: 0.5).
hjust.y	Horizontal justification for the text on the y-axis (default: 1).
hjust.legend	Horizontal justification for the text in the legend title (default: 0.5).
size.x	Text size (in pts) on the x-axis (default: 8).
size.y	Text size (in pts) on the y-axis (default: 10).
legend.position	Position of the legend. Options are "top", "bottom", "right" (default), and "left"; can be abbreviated.
legend.direction	Direction of the legend. Options are "vertical" (default) and "horizontal"; can be abbreviated.
...	Additional arguments (ignored).

Value

A ggplot object.

Examples

```
# Locate file "mvnma_examples.rda" with mvnma() results
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

# Rank the treatments
ranks_sucra <- mvrank(mvnma_all,
  small.values = c("undes", "undes", "des", "des", "des"),
  method = "SUCRA")
ranks_sucra

# Create a heatplot sorting the results according to the first outcome
# (default), i.e., Early_Response
heatplot(ranks_sucra)

# Create a heatplot sorting the results according to the second outcome
```

```
# appearing in ranks_sucra (i.e Early_Remission)
heatplot(ranks_sucra, sort = 2)

# Create a heatplot sorting the results by explicitly mentioning the
# name of the outcome Early_Remission
heatplot(ranks_sucra, sort = "Early_Remission")
```

`linechart` *Line chart showing the VIKOR results*

Description

A line chart showing the results of VišeKriterijumska Optimizacija I Kompromisno Rešenje (VIKOR) across the three metrics Q, S and R for each treatment. Within each metric, lower values represent better treatment performance.

Usage

```
linechart(
  x,
  sort = "Q",
  exclude = "none",
  n = nrow(x),
  linewidth = 1.1,
  size = 2,
  ...
)
```

Arguments

`x`	An object of class `vikor.mvrank`.
`sort`	A character specifying the order of treatments on the x-axis. By default, the order is according to the Q-metric (`"Q"`). Ordering treatments according to the S (`"S"`) and R (`"R"`) metrics is also possible.
`exclude`	A character specifying a metric that will not be displayed in the graph. By default, all metrics are displayed (`"none"`). Alternative options are to exclude either the Q-metric (`"Q"`), the S-metric (`"S"`), or the R-metric (`"R"`).
`n`	A numeric value indicating the number of treatments to be plotted in the graph. By default, all treatments are displayed. If specified, only the first `"n"` treatments according to the hierarchy specified from `"sort"` argument are plotted.
`linewidth`	A numeric value specifying the width of the lines (default: 1.1).
`size`	A numeric value specifying the size of the points (default: 2).
`...`	Additional arguments passed to `ggplot` function.

Value

A ggplot object.

Examples

```
# Locate file "mvnma_examples.rda" with mvnma() results
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

# Rank treatments using SUCRAs (default)
ranks12 <- mvrank(mvnma12, small.values = c("undes", "undes"))
ranks12

# Get the best compromise solution across the efficacy outcomes
vk12 <- vikor(ranks12)

# Visualize the results with default settings
linechart(vk12)

# Sort by the "R" metric
linechart(vk12, sort = "R")

# Sort by the "R" metric and include only the first 3 treatments
linechart(vk12, sort = "R", n = 3)

# Exclude the "R" metric
#
linechart(vk12, exclude = "R")
```

---

mvnma *Perform a Bayesian multivariate network meta-analysis using a single-correlation coefficient model*

Description

This function fits a Bayesian multivariate network meta-analysis model for two or more outcomes. It can also be used to fit a univariate network meta-analysis model. Additionally, the studies can have multiple arms.

Usage

```
mvnma(
  ...,
  method = "standard",
  pooled = "random",
  n.domain = NULL,
  reference.group = NULL,
  outclab = NULL,
```

```
  n.chains = 4,
  n.iter = 10000,
  n.burnin = 2000,
  n.thin = max(1, floor((n.iter - n.burnin)/1000)),
  level = gs("level.ma"),
  scale.psi,
  psi.preset = NULL,
  lower.rho,
  upper.rho,
  varTE.missing = NULL,
  quiet = FALSE
)

## S3 method for class 'mvnma'
print(
  x,
  digits = gs("digits"),
  digits.sd = gs("digits.sd"),
  print.sd = FALSE,
  ...
)
```

Arguments

`...`	Additional arguments (ignored)
`method`	A character string specifying the method to be used for model fitting. This can be either "standard" (default), referring to the standard model, or "DM", referring to the DuMouchel model. The argument can be abbreviated. Set to "standard" for a single outcome.
`pooled`	A character string indicating whether the common effects (`"common"`) or random effects (`"random"`, default) model should be considered. Can be abbreviated.
`n.domain`	Integer indicating the position of the last outcome in the first outcome domain (based on the order of the supplied pairwise objects). Used with ‘method = "DM"‘ to restrict information sharing within outcome domains. Ignored when ‘method = "standard"‘. Default is ‘NULL‘.
`reference.group`	A common reference treatment across all outcomes.
`outclab`	An optional argument with labels for each outcome. If NULL, outcomes are labelled as ’outcome_1’, ’outcome_2’, etc.
`n.chains`	Number of Markov chains (default=4).
`n.iter`	Number of iterations (default: 10000).
`n.burnin`	Number of iterations for burn-in (default: 2000).
`n.thin`	Thinning rate. Default is equal to `max(1, floor((n.iter - n.burnin) / 1000))`.
`level`	The level used to calculate confidence intervals for network estimates.
`scale.psi`	Values for the scale parameter(s) of the half-normal prior used for the heterogeneity parameters within each outcome. If NULL, all values are set to 1. If specified, it should have a length equal to the number of outcomes.

psi.preset	Prespecified value for the between-study heterogeneity parameter (default: 'NULL'). If specified, it must be a numeric vector of length equal to the number of outcomes, which will overwrite arguments 'scale.psi' and 'pooled'.
lower.rho	Lower bounds for the uniform prior(s) used for the correlation coefficient. If NULL, all bounds are set to -1. Input to this argument is ignored for a single outcome.
upper.rho	Upper bounds for the uniform prior(s) used for the correlation coefficient. If NULL, all bounds are set to 1. Input to this argument is ignored for a single outcome.
varTE.missing	Assumed (very large) variance for outcomes not reported in a study. By default, the largest variance times 1000000 is used. This is the same value used for argument seTE.ignore in netimpact to mimicking the removal of individual studies from the network meta-analysis.
quiet	A logical indicating whether to print information on the progress of the JAGS model fitting.
x	An object of class mvnma.
digits	Minimal number of significant digits, see print.default.
digits.sd	Minimal number of significant digits for standard deviations
print.sd	A logical specifying whether standard deviations should be printed.

Details

The multivariate network meta-analysis (mvNMA) model supported by this package refers to the single correlation coefficient model, interpreted as an amalgam of within- and across-outcome correlations (Efthimiou et al., 2015) which is a generalisation of Riley et al. (2008).

The function mvnma expects one or more outcomes / pairwise objects. A common reference treatment across all outcomes is required to only show comparisons with the reference in forest plots. For a single pairwise object, a univariate network meta-analysis is conducted.

The Bayesian multivariate network meta-analysis model fitted in the **mvnma** package assumes uniform priors for the between-outcome correlation coefficients. The lower and upper bounds of these priors can be defined using the arguments 'lower.rho' and 'upper.rho'. If not set, the model will assume a 'Unif (-1, 1)' prior for all correlation coefficients. For two outcomes, a single value can be provided for 'lower.rho' and 'upper.rho'. For example, 'lower.rho' = 0.5 and 'upper.rho' = 1 for rho12 ~ Unif (0.5, 1)). For more than two outcomes, the order in which the bounds are provided matters. For example, when pooling four outcomes, the lower and upper bounds correspond to the following order of correlation coefficients: (rho12, rho13, rho14, rho23, rho24, rho34).

Two types of priors for the treatment effect parameters are supported via the argument 'method'. Setting 'method = "standard"' fits an mvNMA model using non-informative normal priors (e.g., 'N(0, 10^3)').

Alternatively, 'method = "DM"' specifies the DuMouchel prior, which assumes constant relative treatment effects across outcomes and enables information sharing (DuMouchel & Harris, 1983). This may improve precision but can introduce bias when outcomes from different domains (e.g., efficacy and safety) are analyzed jointly. Note, the DuMouchel model is not available for a single outcome.

The argument ‘n.domain‘ can be used to restrict information sharing to predefined outcome domains. It indicates the position (based on the order of the supplied pairwise objects) of the last outcome in the first domain. For example, with four outcomes, setting ‘n.domain = 2‘ assigns the first two outcomes to one domain and the remaining outcomes to a second domain. In this case, information is shared only within domains.

By default, ‘n.domain = NULL‘, in which case information is shared across all outcomes when ‘method = "DM"‘. This may be appropriate when all outcomes belong to the same domain or when cross-domain sharing is justified.

The argument ‘n.domain‘ is ignored when ‘method = "standard"‘.

Value

The function returns an ’mvnma’ object. This consists of the results for each outcome and, for more than one outcome, the correlation coefficient estimates between the combined outcomes. The outcome-specific estimates are expressed in the format of a list (one for each outcome) which contains:

- The basic estimates (i.e. treatment vs. reference.group) for each outcome.
- The heterogeneity estimates for each outcome
- The posterior samples corresponding to the basic estimates.

References

DuMouchel WH, Harris JE (1983): Bayes methods for combining the results of cancer studies in humans and other species. *Journal of the American Statistical Association*, **78**, 293–308

Efthimiou O, Mavridis D, Riley RD, Cipriani A, Salanti G (2015): Joint synthesis of multiple correlated outcomes in networks of interventions. *Biostatistics*, **16**, 84–97

Riley RD, Thompson JR, Abrams KR (2008): An alternative model for bivariate random-effects meta-analysis when the within-study correlations are unknown. *Biostatistics*, **9**, 172–86

See Also

pairwise

Examples

```
# Use 'pairwise' to obtain contrast based data for the first two outcomes

# Early response
pw1 <- pairwise(treat = list(treatment1, treatment2, treatment3),
  event = list(resp1, resp2, resp3), n = list(n1, n2, n3),
  studlab = id, data = Linde2015, sm = "OR")

# Early remissions
pw2 <- pairwise(treat = list(treatment1, treatment2, treatment3),
  event = list(remi1, remi2, remi3), n = list(n1, n2, n3),
  studlab = id, data = Linde2015, sm = "OR")

# Define outcome labels
```

```
outcomes <- c("Early_Response", "Early_Remission",
  "Adverse_events", "Loss_to_follow_up", "Loss_to_follow_up_AE")

# Fit the model combining only the two efficacy outcomes
# (note, we are using only 10 iterations and 2 burnins to reduce the
#  runtime of the example; in real applications use larger numbers)
set.seed(1910)
mvnma(pw1, pw2,
  reference.group = "Placebo", outclab = outcomes[1:2],
  n.iter = 10, n.burnin = 2)


# Use 'pairwise' to obtain contrast based data for the third to fifth
# outcome

# Adverse events
pw3 <- pairwise(treat = list(treatment1, treatment2,treatment3),
  event = list(ae1, ae2, ae3),  n = list(n1, n2, n3),
  studlab = id, data = Linde2015, sm = "OR")

# Loss to follow-up
pw4 <- pairwise(treat = list(treatment1, treatment2, treatment3),
  event = list(loss1, loss2, loss3), n = list(n1, n2, n3),
  studlab = id, data = Linde2015, sm = "OR")

# Loss_to_follow_up_(AE)
pw5 <- pairwise(treat = list(treatment1, treatment2, treatment3),
  event = list(loss.ae1, loss.ae2, loss.ae3), n = list(n1, n2, n3),
  studlab = id, data = Linde2015, sm = "OR")

# Fit the model combining only the two efficacy outcomes
# (note, we are using only 100 iterations and 20 burnins to reduce the
#  runtime of the example; in real applications use larger numbers)
set.seed(1909)
mvnma12 <- mvnma(pw1, pw2,
  reference.group = "Placebo", outclab = outcomes[1:2],
  n.iter = 100, n.burnin = 20)
mvnma12

# Extract treatment effect estimates and heterogeneity for Early_Response
mvnma12$Early_Response$basic_estimates
mvnma12$Early_Response$heterogeneity

# Extract outcome correlation
mvnma12$cor

# Plot the results for efficacy outcomes
forest(mvnma12)

# Print odds ratios for efficacy outcomes
outc <- names(mvnma12)[names(mvnma12) != "cor"]
#
for (i in outc) {
```

```
  cat(paste0("\nOutcome: ", i, "\n\n"))
  print(round(exp(mvnma12[[i]]$TE.random), 2))
}

# Fit the model combining all five outcomes
# (note, we are using only 100 iterations and 20 burnins to reduce the
#  runtime of the example; in real applications use larger numbers)
set.seed(1904)
mvnma_all <- mvnma(pw1, pw2, pw3, pw4, pw5,
  reference.group = "Placebo", outclab = outcomes,
  n.iter = 100, n.burnin = 20)

# Extract treatment effect estimates and heterogeneity for Early_Response
mvnma_all$Early_Response$basic_estimates
mvnma_all$Early_Response$heterogeneity

# Extract outcome correlation
mvnma_all$cor

# Plot the results for all outcomes
forest(mvnma_all)

# Print odds ratios for all outcomes
outc <- names(mvnma_all)[names(mvnma_all) != "cor"]
#
for (i in outc) {
  cat(paste0("\nOutcome: ", i, "\n\n"))
  print(round(exp(mvnma_all[[i]]$TE.random), 2))
}

# Results for a univariate Bayesian network meta-analysis
set.seed(1941)
bnma1 <- mvnma(pw1,
  reference.group = "Placebo", outclab = outcomes[1],
  n.iter = 100, n.burnin = 20)
bnma1

# Results for a univariate frequentist network meta-analysis
nma1 <- netmeta(pw1, reference.group = "Placebo", common = FALSE)
print(nma1, backtransf = FALSE)
```

---

mvrank *Outcome-specific treatment rankings in multivariate network meta-analysis.*

Description

Produces outcome-specific treatment rankings in multivariate network meta-analysis based on the output of the mvnma function.

This function produces outcome-specific treatment rankings in multivariate network meta-analysis based on the output of the mvnma function. Two ranking methods (argument method) are currently supported (Salanti et al., 2011):

- Surface under the cumulative ranking curve (SUCRA) method,
- Probability of being best (pbest) method.

Usage

```
mvrank(x, small.values, method = "SUCRA")

## S3 method for class 'mvrank'
print(x, digits = gs("digits"), ...)
```

Arguments

x	An object of class mvnma.
small.values	A character vector specifying for each outcome whether small treatment effects indicate a beneficial ("desirable") or harmful ("undesirable") effect, can be abbreviated.
method	The ranking method to be used. Three methods are currently supported. The SUCRA method (specified as method = "SUCRA") is the default approach. The probability of being best method (specified as method = "pbest") and the mean and median ranks (specified as method = "ranks") are also supported.
digits	Minimal number of significant digits, see print.default.
...	Additional arguments (ignored)

Value

The function returns an 'mvrank' object which is a list consisting of a data frame with the variables 'treatment' and either 'SUCRA' or 'pbest' for each outcome in the multivariate network meta-analysis.

References

Salanti G, Ades AE, Ioannidis JP (2011): Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *Journal of Clinical Epidemiology*, **64**, 163–71

Examples

```
# Locate file "mvnma_examples.rda" with mvnma() results
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

# Rank treatments using SUCRAs (default)
ranks_sucra <- mvrank(mvnma_all,
  small.values = c("undes", "undes", "des", "des", "des"))
#
```

```
ranks_sucra


# Rank treatments using pbest
ranks_pbest <- mvrank(mvnma_all,
  small.values = c("undes", "undes", "des", "des", "des"),
  method = "pbest")
#
ranks_pbest

# Rank treatments using mean and median ranks
ranks_mean_median <- mvrank(mvnma_all,
  small.values = c("undes", "undes", "des", "des", "des"),
  method = "ranks")
#
ranks_mean_median
```

`netsplit.mvnma` *Split direct and indirect evidence in multivariate network meta-analysis*

Description

This function performs local inconsistency checks using the node-splitting method.

Usage

```
## S3 method for class 'mvnma'
netsplit(
  x,
  univariate = TRUE,
  tol.direct = 5e-04,
  seed = NULL,
  quiet = TRUE,
  ...
)

## S3 method for class 'netsplit.mvnma'
print(
  x,
  backtransf = gs("backtransf"),
  digits = gs("digits"),
  digits.se = gs("digits.se"),
  digits.pval = gs("digits.pval"),
  digits.stat = gs("digits.stat"),
  digits.prop = max(gs("digits.pval") - 2, 2),
```

```
  print.se = FALSE,
  ci = TRUE,
  k = TRUE,
  prop = TRUE,
  overall = TRUE,
  direct = TRUE,
  indirect = TRUE,
  diff = TRUE,
  z = TRUE,
  test = TRUE,
  scientific.pval = gs("scientific.pval"),
  text.NA = gs("lab.NA"),
  legend = TRUE,
  ...
)
```

Arguments

x	An object of class mvnma.
univariate	A logical indicating how direct estimates are obtained, either from univariate, outcome-specific, pairwise meta-analysis (univariate = TRUE, default) or from multivariate network meta-analyses. Under the random effects model, the between-study heterogeneity is taken from the object x for both cases.
tol.direct	A numeric defining the maximum deviation of the direct evidence proportion from 0 or 1 to classify a comparison as providing only indirect or direct evidence, respectively. No indirect estimate is reported for such comparisons.
seed	An optional numeric value used to set the random number generator before fitting the node-splitting models, in order to obtain reproducible results. Only relevant for univarite = FALSE, where direct estimates are obtained from an MCMC model fit. The state of the random number generator is restored on exit.
quiet	A logical indicating whether to print information on the progress of the JAGS model fitting. Only relevant for univarite = FALSE, where direct estimates are obtained from an MCMC model fit.
...	Additional arguments (ignored).
backtransf	A logical indicating whether results should be back transformed for printing. For example, if backtransf = TRUE, results for sm = "OR" are printed as odds ratios rather than log odds ratios, and the difference between the direct and indirect estimate is printed as a ratio of ratios. Estimates are always stored on the original scale in the netsplit object; only the printed output is affected. Standard errors and z-values are never back transformed.
digits	Minimal number of significant digits for treatment estimates and confidence intervals, see print.default.
digits.se	Minimal number of significant digits for standard errors, see print.default.
digits.pval	Minimal number of significant digits for p-values of the test for disagreement between direct and indirect evidence, see print.default.

digits.stat	Minimal number of significant digits for z-values of the test for disagreement between direct and indirect evidence, see print.default.
digits.prop	Minimal number of significant digits for direct evidence proportions, see print.default.
print.se	A logical indicating whether standard errors should be printed.
ci	A logical indicating whether confidence intervals should be printed.
k	A logical indicating whether the number of studies providing direct evidence should be printed.
prop	A logical indicating whether the direct evidence proportion should be printed.
overall	A logical indicating whether estimates from the multivariate network meta-analysis should be printed.
direct	A logical indicating whether estimates derived from direct evidence should be printed.
indirect	A logical indicating whether estimates derived from indirect evidence should be printed.
diff	A logical indicating whether the difference between direct and indirect treatment estimates should be printed.
z	A logical indicating whether z-values of the test for disagreement between direct and indirect evidence should be printed.
test	A logical indicating whether p-values of the test for disagreement between direct and indirect evidence should be printed.
scientific.pval	A logical specifying whether p-values should be printed in scientific notation, e.g., 1.2345e-01 instead of 0.12345.
text.NA	A character string specifying text printed for missing values.
legend	A logical indicating whether a legend should be printed.

Details

Node-splitting is applied to every treatment comparison that is informed by both direct and indirect evidence. A comparison qualifies if at least one study reports it for the outcome and if the two treatments remain connected in the network once that comparison is removed. Only the comparison itself is removed, not the studies providing it, so that the remaining arms of a multi-arm study still contribute to the indirect estimate (Dias et al., 2010). The set of comparisons that can be split may differ between outcomes, as an outcome is typically not reported by all studies in the network.

For each qualifying comparison, the direct estimate is obtained from the studies directly comparing the two treatments, either by an outcome-specific pairwise meta-analysis or by re-fitting the multivariate model to the direct evidence (see argument univariate). The indirect estimate is then derived by back-calculation from the multivariate network meta-analysis estimate and the direct estimate, using the direct evidence proportion, that is, the ratio of the network variance to the direct variance.

Back-calculation divides both the indirect estimate and its variance by one minus the direct evidence proportion. Results therefore become unstable as the proportion approaches 1, and estimates for such comparisons should be interpreted with caution. No indirect estimate is reported if the proportion is not between tol.direct and 1 - tol.direct; this includes proportions above 1, which can occur when the network estimate is less precise than the direct estimate.

Estimates are stored on the original scale and back transformed only for printing (see argument backtransf). For ratio measures, the difference between the direct and indirect estimate is a ratio of ratios and is printed as RoR. Standard errors, z-values and p-values always refer to the original scale.

Value

An object of class netsplit.mvnma; a list with one data frame per outcome, with one row per treatment comparison and the following columns:

comparison	Treatment comparison.
k	Number of studies providing direct evidence.
prop	Direct evidence proportion. Reported even when it falls outside the range for which an indirect estimate is calculated.
mvnma.TE, mvnma.seTE, mvnma.lb, mvnma.ub	Estimated treatment effect, standard error and confidence limits in the multivariate network meta-analysis.
direct.TE, direct.seTE, direct.lb, direct.ub	Estimated treatment effect, standard error and confidence limits derived from direct evidence.
indirect.TE, indirect.seTE, indirect.lb, indirect.ub	Estimated treatment effect, standard error and confidence limits derived from indirect evidence.
diff, se.diff, diff.lb, diff.ub	Difference between direct and indirect estimates, its standard error and confidence limits.
z, p.val	z-value and p-value of the test for disagreement between direct and indirect evidence.
sign	A logical indicating whether the disagreement is statistically significant.

All estimates are on the original scale, that is, not back transformed (see argument backtransf). Outcomes without any treatment comparison contributing both direct and indirect evidence are set to NULL, and NULL is returned with a warning if no such comparison exists for any outcome.

References

Dias S, Welton NJ, Caldwell DM, Ades AE (2010): Checking consistency in mixed treatment comparison meta-analysis. *Statistics in Medicine*, **29**, 932–44

König J, Krahn U, Binder H (2013): Visualizing the flow of evidence in network meta-analysis and characterizing mixed treatment comparisons *Statistics in Medicine*, **32**, 5414–29

Rücker G, Nikolakopoulou A, Papakonstantinou T, Salanti G, Riley RD, Schwarzer G (2020): The statistical importance of a study for a network meta-analysis estimate. *BMC Medical Research Methodology*, **20**, 190

Examples

```
# Locate file "mvnma_examples.rda" with mvnma() results
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

# Local checks for inconsistency
print(netsplit(mvnma12), backtransf = FALSE)
```

`plot.mvrank` *Scatter plot to visualize the ranking of two outcomes in a multivariate network meta-analysis*

Description

Draw a scatter plot in the active graphics window.

Usage

```
## S3 method for class 'mvrank'
plot(
  x,
  which = 1:2,
  pos = 1,
  cex.point = 1,
  cex.label = 0.7,
  pch = 19,
  xlim = c(0, 1),
  ylim = c(0, 1),
  ...
)
```

Arguments

`x`	An object of class `mvrank`.
`which`	A mandatory numeric vector of length 2 specifying which outcomes should be plotted. For example, setting "outcome = c(2, 3)" implies that a scatter plot will be generated plotting the rankings of outcomes 2 and 3.
`pos`	Position of treatment labels.
`cex.point`	a numeric value specifying the size of the points (default: 1)
`cex.label`	a numeric value specifying the size of the point labels in the plot (default: 0.7)
`pch`	a vector of plotting characters or symbols (default: 19)
`xlim`	the x limits of the plot
`ylim`	the y limits of the plot
`...`	Additional arguments for `plot` function.

Value

A scatter plot is shown in the active graphics window.

Examples

```
# Locate file "mvnma_examples.rda" with mvnma() results
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

# Rank treatments using SUCRAs (default)
ranks12 <- mvrank(mvnma12, small.values = c("undes", "undes"))
ranks12

# Visualize SUCRAs in a scatter plot with outcome 1 on the x-axis and
# outcome 2 on the y-axis
plot(ranks12)

# Visualize SUCRAs in a scatter plot with outcome 2 on the x-axis and
# outcome 1 on the y-axis
plot(ranks12, which = 2:1)
```

---

spiechart.mvrank *Rank treatments across all outcomes using the spie chart method*

Description

This function employs the spie chart approach to combine outcome-specific treatment hierarchies obtained in terms of the probabilistic ranking metrics:

- Surface Under the Cumulative Ranking curve (SUCRA), or
- probability of being best (pbest).

Usage

```
## S3 method for class 'mvrank'
spiechart(x, weights = NULL, ...)

spiechart(x, ...)
```

Arguments

x	An object of class mvrank.
weights	Outcome weights. The weights should always sum to 1. If not then they are standardized. If NULL, the function will assume equal outcome weights.
...	Additional arguments passed to ggplot function.

Details

The spie chart is a modified pie chart in which each sector corresponds to a treatment and its radius is proportional to a ranking metric, so that sector area encodes treatment performance (Daly et al., 2020). This function constructs one spie chart per outcome using SUCRA or probability of being best (pbest) metrics, where each metric determines the radius of the corresponding sector.

An amalgamated treatment hierarchy across outcomes is then derived by averaging the sector areas across outcomes, providing a quantitative summary of overall treatment performance.

Value

A data frame with the area under the spie chart as amalgamated treatment hierarchy across outcomes.

References

Daly CH, Mbuagbaw L, Thabane L, Straus SE, Hamid JS (2020): Spie charts for quantifying treatment effectiveness and safety in multiple outcome network meta-analysis: a proof-of-concept study. *BMC Med Res Methodol*, **20**, 266

Examples

```
# Locate file "mvnma_examples.rda" with mvnma() results
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

# Rank treatments using SUCRAs
ranks12 <- mvrank(mvnma12, method = "sucra",
  small.values = c("undes", "undes"))
ranks12

# Get amalgamated treatment hierarchy
spiechart(ranks12)
```

---

vikor.mvrank *Rank treatments across all outcomes using the VišeKriterijumska Optimizacija I Kompromisno Rešenje (VIKOR) multi-criteria decision analysis method*

Description

This function employs the VišeKriterijumska Optimizacija I Kompromisno Rešenje (VIKOR) method to analyze all outcome-specific ranking lists. It provides both an amalgamated ranking list and guidance on which treatments correspond to the best compromise solutions.

Usage

```
## S3 method for class 'mvrank'
vikor(x, weights = NULL, v = 0.5, ...)
```

Arguments

x	An object of class mvrank or a matrix.
weights	Outcome weights. The weights should always sum to 1. If not then they are standardized. If NULL, the function will assume equal outcome weights.
v	A scalar from 0 to 1 interpreted as the weight of the decision making process. Following guidance from the multi-criteria decision analysis field it is set to 0.5.
...	Additional arguments (ignored).

Details

This function takes a single mandatory argument, which is either an object of class mvrank or a matrix. It then uses the multi-criteria decision analysis method VišeKriterijumska Optimizacija I Kompromisno Rešenje (VIKOR) to produce an amalgamated ranking list across all outcomes (Opricovic & Tzeng, 2004).

The standard VIKOR approach is applied when the method argument is set to "sucra" or "pbest" in mvrank.

A fuzzy VIKOR method (Opricovic, 2011) is applied when outcome-specific rankings are expressed in terms of median ranks and 95% credible intervals. The latter is possible when the mvrank object is created with method = "ranks".

In both cases, the final ranking list is calculated based on treatments common across all outcomes. Treatments not present across all outcomes are excluded internally.

Using the argument 'weights' the users can specify the weight that each outcome should have in the decision making process. For each outcome this argument should have a value from 0 to 1 while the sum of all outcome weights should be 1. If the sum of all weights is not 1, then these are internally standardize to achieve this. The standardized weight values are returned as a message to the user. Finally, if NULL then equal weights are assumed across all outcomes.

The argument 'v' specifies the weight of the decision making process. The VIKOR method is a compromise programming approach that aims to balance between each treatments overall and worst performance across all outcomes. The balance between these two criteria is achieved using the parameter 'v' which takes values from 0 to 1. Values close to 1 will give more weight to the treatment's overall performance while values close to 0 will give more weight to penalize the treatment's worst performance. The most common choice of 'v' is typically 0.5 (default also here), thereby allowing for a balanced decision making between treatment's overall and worst performance.

Value

The function returns a 'vikor' object. This consists of three ranking lists which are the following:

- A ranking list Q referring to the ranking when balancing both each treatment's overall and worst performance. This is the main ranking list of the method.
- A ranking list S referring to the ranking in terms of each treatment's overall performance.
- A ranking list R referring to the ranking in terms of penalising each treatment's worst performance.

In addition to the ranking lists, the function also evaluates the necessary conditions defined by the VIKOR method and returns a message indicating the set of compromise solutions.

References

Opricovic S, Tzeng GH (2004): Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European Journal of Operational Research*, **156**, 445–55

Opricovic S (2011): Fuzzy VIKOR with an application to water resources planning. *Expert Systems with Applications*, **38**, 12983–90

See Also

print.vikor

Examples

```
# Locate file "mvnma_examples.rda" with mvnma() results
.fname <- system.file("extdata/mvnma_examples.rda", package = "mvnma")
load(.fname)

# Print the results of a bivariate network meta-analysis
mvnma12

# Rank treatments using SUCRAs
ranks12 <- mvrank(mvnma12, method = "sucra",
  small.values = c("undes", "undes"))
ranks12

# Get the best compromise solution across the efficacy outcomes
vikor(ranks12)

# Use larger weight for response than remission
vikor(ranks12, weights = c(0.6, 0.3))
```

Index

∗ **hplot**
    forest.mvnma, 6
    plot.mvrank, 26
∗ **package**
    mvnma-package, 2

as.mcmc *(*as.mcmc.mvnma*)*, 5
as.mcmc.list, *5*
as.mcmc.mvnma, *3*, 5

forest.meta, *7*, *10*
forest.mvnma, *3*, 6
forest.netsplit.mvnma, *3*, 8

ggplot, *14*, *27*

hasse.mvrank, *3*, 11
hasse.netposet, *11*
heatplot.mvrank, *3*, 12

linechart, *3*, 14

mvnma, *3*, *5*, *6*, 15, *17*, *20*, *21*, *23*
mvnma-package, 2
mvrank, *3*, *11–13*, 20, *26*, *27*, *29*

netimpact, *17*
netsplit.mvnma, *3*, *10*, 22

pairwise, *17*, *18*
plot, *26*
plot.mvrank, *3*, 26
print.mvnma *(*mvnma*)*, 15
print.mvrank *(*mvrank*)*, 20
print.netsplit.mvnma *(*netsplit.mvnma*)*, 22
print.vikor, *30*

spiechart, *3*
spiechart *(*spiechart.mvrank*)*, 27
spiechart.mvrank, 27

vikor.mvrank, *3*, *14*, 28