

Package ‘mvnma’

May 19, 2026

Title Multivariate Network Meta-Analysis using Bayesian Methods

Version 0.1-0

Date 2026-05-15

Depends R (>= 4.0.0), meta (>= 8.3-0), netmeta (>= 3.4-0)

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](https://doi.org/10.1093/biostatistics/kxu030)>;
- per-outcome treatment hierarchies using the surface under the cumulative ranking curve (SU-CRA), the probability of best value, or median (or mean) ranks (Salanti et al., 2011) <[doi:10.1016/j.jclinepi.2010.03.016](https://doi.org/10.1016/j.jclinepi.2010.03.016)>;
- across-outcomes benefit-risk assessment using the VišeKriterijumska Optimizacija I Kompromisno Rešenje (VIKOR) method (Opricovic & Tzeng, 2004) <[doi:10.1016/S0377-2217\(03\)00020-1](https://doi.org/10.1016/S0377-2217(03)00020-1)>;
- convergence checks using trace plots, density plots, or the R-hat statistic;
- forest plots of treatment estimates, scatter plots of per-outcome rankings, Hasse diagrams (Carlsen & Bruggemann, 2014) <[doi:10.1002/cem.2569](https://doi.org/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 8.0.0

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 CRAN

Date/Publication 2026-05-19 09:50:02 UTC

Contents

mvnma-package	2
as.mcmc.mvnma	4
forest.mvnma	5
hasse.mvrank	7
heatplot.mvrank	8
linechart	10
mvnma	11
mvrank	16
plot.mvrank	18
spiechart.mvrank	19
vikor.mvrank	20
Index	23

 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.

DuMouchel priors assuming constant relative treatment effects across outcomes and enabling information sharing can be used (DuMouchel & Harris, 1983). This may improve precision but can introduce bias when outcomes from different domains (e.g., efficacy and safety) are analyzed jointly.

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 possible using ranking methods such as the surface under the cumulative ranking curve (SUCRA) (Salanti et al., 2011), the probability of best value, 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 provide an amalgamated treatment hierarchy across outcomes and 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 checked 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; 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 to yield 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 `mvrnk` to get outcome-specific treatment rankings.
- Function `vikor` 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 `plot.mvrnk` to visualize per outcome ranking results for any pair of outcomes.
- Function `hasse.mvrnk` to visualize the partial order of the treatment across all outcomes.
- Function `heatplot.mvrnk` to visualize in a heatmap 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 **mvnma** 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

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

<code>x</code>	An object of class <code>mvnma</code> .
<code>backtransf</code>	A logical indicating whether results should be back transformed in forest plots. If <code>backtransf = TRUE</code> , results for <code>sm = "OR"</code> are presented as odds ratios rather than log odds ratios, for example.
<code>separate</code>	A logical indicating whether separate forest plots should be created for outcomes.
<code>leftcols</code>	A character vector specifying (additional) columns to be printed on the left side of the forest plot or a logical value.
<code>leftlabs</code>	A character vector specifying labels for (additional) columns on left side of the forest plot.
<code>rightcols</code>	A character vector specifying (additional) columns to be printed on the right side of the forest plot or a logical value.
<code>rightlabs</code>	A character vector specifying labels for (additional) columns on right side of the forest plot.
<code>col.study</code>	The colour for treatment effect estimates and confidence interval limits.
<code>col.square</code>	The colour for squares reflecting the treatment effect estimates.
<code>col.square.lines</code>	The colour for the outer lines of squares reflecting the treatment effect estimates.
<code>col.subgroup</code>	The colour to print information on different outcomes.
<code>squaresize</code>	The size of squares reflecting the treatment effect estimates (default: 0.7).
<code>header.line</code>	A logical value indicating whether to print a header line (default: TRUE) or a character string ("both", "below", "").
<code>text.outcome</code>	A text string printed before the name of the outcome.
<code>...</code>	Additional arguments passed on to <code>forest.meta</code> .

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)
```

hasse.mvrank	<i>Hasse diagram</i>
--------------	----------------------

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 heatmap to visualize the ranking in a multivariate network meta-analysis

Description

This function produces a heatmap 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 best value (pBV).

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

<code>x</code>	An object of class <code>mvrank</code> .
<code>sort</code>	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 <code>mvrank</code> , is used as the reference. The user can either provide a single number corresponding to the position of the outcome in the <code>mvrank</code> object or a character string with the name of the outcome; can be abbreviated.
<code>col.num</code>	The color of the numbers in the squares (default: "white").
<code>num.size</code>	The size of the numbers in the squares (default: 4.5).
<code>col.low</code>	The color for the low end of the gradient (default: "lightblue").
<code>col.high</code>	The color for the high end of the gradient (default: "darkblue").
<code>width.bar</code>	The width of the bars in the graph (default: 1.2).
<code>angle.x</code>	The angle (in [0, 360]) to rotate the text appearing on the x-axis (default: 0).
<code>angle.y</code>	The angle (in [0, 360]) to rotate the text appearing on the y-axis (default: 0).
<code>hjust.x</code>	Horizontal justification for the text on the x-axis (default: 0.5).
<code>hjust.y</code>	Horizontal justification for the text on the y-axis (default: 1).
<code>hjust.legend</code>	Horizontal justification for the text in the legend title (default: 0.5).
<code>size.x</code>	Text size (in pts) on the x-axis (default: 8).
<code>size.y</code>	Text size (in pts) on the y-axis (default: 10).
<code>legend.position</code>	Position of the legend. Options are "top", "bottom", "right" (default), and "left"; can be abbreviated.
<code>legend.direction</code>	Direction of the legend. Options are "vertical" (default) and "horizontal"; can be abbreviated.
<code>...</code>	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 <- mvrnk(mvnma_all,
  small.values = c("undes", "undes", "des", "des", "des"),
  method = "SUCRA")
ranks_sucra

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

# Create a heatmap sorting the results according to the second outcome
# appearing in ranks_sucra (i.e Early_Remission)
heatmap(ranks_sucra, sort = 2)

# Create a heatmap sorting the results by explicitly mentioning the
# name of the outcome Early_Remission
heatmap(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 <code>vikor</code> .
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 <code>ggplot</code> 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 <- mvrnk(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. Currently, the function can simultaneously pool up to five outcomes. Additionally, the studies to be included should be of maximum three arms.

Usage

```

mvnma(
  ...,
  reference.group = NULL,
  outclab = NULL,
  n.domain = 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,
  lower.rho,
  upper.rho,
  method = "standard",
  quiet = FALSE
)

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

```

Arguments

...	Additional arguments (ignored)
reference.group	A common reference treatment across all outcomes.
outclab	An optional argument with labels for each outcome. If NULL, the each outcome is labelled as 'outcome_1', 'outcome_2' etc.
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'.
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 $\max(1, \text{floor}((n.iter - n.burnin) / 1000))$.
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.

lower.rho	Lower bounds for the Uniform prior(s) used for the correlation coefficient. If NULL all bounds are set to -1.
upper.rho	Upper bounds for the Uniform prior(s) used for the correlation coefficient. If NULL all bounds are set to 1.
method	A character string specifying the method to be used for model fitting. This can be either "standard" (default), referring to the standard bivariate model, or "DM", referring to the bivariate model based on the DuMouchel method. The argument can be abbreviated.
quiet	A logical indicating whether to print information on the progress of the JAGS model fitting.
x	An object of class <code>mvnma</code> .
digits	Minimal number of significant digits, see <code>print.default</code> .
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 two to five outcomes / `pairwise` objects. A common reference treatment across all outcomes is required to only show comparisons with the reference in forest plots.

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.

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 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))
}

```

mvrnk

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 best value (pBV) method.

Usage

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

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

Arguments

<code>x</code>	An object of class mvnma .
<code>small.values</code>	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 <code>method = "SUCRA"</code>) is the default approach. The probability of best value method (specified as <code>method = "pBV"</code>) and the mean and median ranks (specified as <code>method = "ranks"</code>) are also supported.
digits	Minimal number of significant digits, see <code>print.default</code> .
...	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 'pBV' 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 pBV
ranks_pBV <- mvrank(mvnma_all,
  small.values = c("undes", "undes", "des", "des", "des"),
  method = "pBV")
#
ranks_pBV

# 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
```

plot.mvrank	<i>Scatter plot to visualize the ranking of two outcomes in a multivariate network meta-analysis</i>
-------------	--

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 <code>mvrank</code> .
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 <code>plot</code> 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 best value (pBV).

Usage

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

spiechart(x, ...)
```

Arguments

x	An object of class <code>mvrank</code> .
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 <code>ggplot</code> 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 best values (pBV) 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, ...)

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

vikor(x, ...)

## S3 method for class 'vikor'
print(x, digits = 4, ...)
```

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).
digits	A numeric specifying the number of digits to print the ranking matrix Q.

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 "pBV" 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 standardized 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

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, 5
plot.mvrank, 18

* **package**

mvnma-package, 2

as.mcmc (as.mcmc.mvnma), 4
as.mcmc.list, 4
as.mcmc.mvnma, 3, 4

forest.meta, 6
forest.mvnma, 3, 5

ggplot, 11, 19

hasse.mvrank, 3, 7
hasse.netposet, 7
heatplot.mvrank, 3, 8

linechart, 3, 10

mvnma, 3, 5, 6, 11, 13, 16
mvnma-package, 2
mvrank, 3, 7–9, 16, 18, 19, 21

pairwise, 13, 14
plot, 18
plot.mvrank, 3, 18
print.mvnma (mvnma), 11
print.mvrank (mvrank), 16
print.vikor (vikor.mvrank), 20

spiechart, 3
spiechart (spiechart.mvrank), 19
spiechart.mvrank, 19

vikor, 3, 10
vikor (vikor.mvrank), 20
vikor.mvrank, 20