Package 'netmeta'

January 2, 2019

Title Network Meta-Analysis using Frequentist Methods

Version 1.0-1

Depends meta (>= 4.9-3)

Suggests colorspace, rgl, hasseDiagram (>= 0.1.3), grid

Imports magic, MASS

Date 2019-01-02

URL https://github.com/guido-s/netmeta http://meta-analysis-with-r.org

Description A comprehensive set of functions providing frequentist methods for network meta-analysis and supporting Schwarzer et al. (2015) <DOI:10.1007/978-3-319-21416-0>, Chapter 8 ``Network Meta-Analysis'':
- frequentist network meta-analysis following Rücker (2012) <DOI:10.1002/jrsm.1058>;
- net heat plot and design-based decomposition of Cochran's Q according to Krahn et al. (2013) <DOI:10.1186/1471-2288-13-35>;
- measures characterizing the flow of evidence between two treatments by König et al. (2013) <DOI:10.1002/sim.6001>;
- ranking of treatments (frequentist analogue of SUCRA) according to Rücker & Schwarzer (2015) <DOI:10.1186/s12874-015-0060-8>;
- partial order of treatment rankings ('poset') and Hasse diagram for 'poset' (Carlsen & Bruggemann, 2014) <DOI:10.1002/cem.2569>; (Rücker & Schwarzer, 2017) <DOI:10.1002/jrsm.1270>;
- split direct and indirect evidence to check consistency (Dias et al., 2010) <DOI:10.1002/sim.3767>;
- league table with network meta-analysis results;
- additive network meta-analysis for combinations of treatments;
- network meta-analysis of binary data using the Mantel-Haenszel or non-central hypergeometric distribution method;
- 'comparison-adjusted' funnel plot (Chaimani & Salanti, 2012) <DOI:10.1002/jrsm.57>;
- automated drawing of network graphs described in Rücker & Schwarzer (2016) <DOI:10.1002/jrsm.1143>.

License GPL (>= 2)

Encoding UTF-8

NeedsCompilation no

Author Gerta Rücker [aut],
Ulrike Krahn [aut],
Jochem König [aut],
Orestis Efthimiou [aut],
Guido Schwarzer [aut, cre]

Maintainer Guido Schwarzer <sc@imbi.uni-freiburg.de>

Repository CRAN

Date/Publication 2019-01-02 14:40:07 UTC

R topics documented:

`netmeta-package` *netmeta: Brief overview of methods and general hints*

Description

R package **netmeta** provides frequentist methods for network meta-analysis and supports Schwarzer et al. (2015), Chapter 8 on network meta-analysis http://meta-analysis-with-r.org/.

Details

R package **netmeta** is an add-on package for **meta** providing the following meta-analysis methods:

- frequentist network meta-analysis (function `netmeta`) based on Rücker (2012) and Rücker & Schwarzer (2014);
- net heat plot (`netheat`) and design-based decomposition of Cochran's Q (`decomp.design`) described in Krahn et al. (2013);
- measures characterizing the flow of evidence between two treatments (`netmeasures`) described in König et al. (2013);
- ranking of treatments (`netrank`) based on frequentist analogue of SUCRA (Rücker & Schwarzer, 2015);
- partial order of treatment rankings (`netposet`, `plot.netposet`) and Hasse diagram (`hasse`) according to Carlsen & Bruggemann (2014) and Rücker & Schwarzer (2017);
- split direct and indirect evidence (`netsplit`) to check for consistency (Dias et al., 2010);
- league table with network meta-analysis results (`netleague`);
- additive network meta-analysis for combinations of treatments (`netcomb`, `discomb` for disconnected networks) (Rücker et al., 2018);
- network meta-analysis of binary data (`netmetabin`) using the Mantel-Haenszel or non-central hypergeometric distribution method (Efthimiou et al., 2018);
- 'comparison-adjusted' funnel plot (`funnel.netmeta`) to assess funnel plot asymmetry in network meta-analysis (Chaimani & Salanti, 2012)
- automated drawing of network graphs (`netgraph`) described in Rücker & Schwarzer (2016);
- results of several network meta-analyses can be combined with `netbind` to show these results in a forest plot.

Furthermore, functions and datasets from **netmeta** are utilised in Schwarzer et al. (2015), Chapter 8 "Network Meta-Analysis", http://meta-analysis-with-r.org/.

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

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

To report problems and bugs

- type `bug.report(package = "netmeta")` if you do not use RStudio,
- send an email to Guido Schwarzer <sc@imbi.uni-freiburg.de> if you use RStudio.

The development version of **netmeta** is available on GitHub https://github.com/guido-s/netmeta.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>, Gerta Rücker <ruecker@imbi.uni-freiburg.de>

References

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

Chaimani A & Salanti G (2012): Using network meta-analysis to evaluate the existence of small-study effects in a network of interventions. Research Synthesis Methods, **3**, 161–76

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

Efthimiou O, Rücker G, Schwarzer G, Higgins J, Egger M, Salanti G (2018): A Mantel-Haenszel model for network meta-analysis of rare events. *Manuscript submitted for publication*.

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

Krahn U, Binder H, König J (2013): A graphical tool for locating inconsistency in network meta-analyses. *BMC Medical Research Methodology*, **13**, 35

Rücker G (2012): Network meta-analysis, electrical networks and graph theory. *Research Synthesis Methods*, **3**, 312–24

Rücker G, Schwarzer G (2014): Reduce dimension or reduce weights? Comparing two approaches to multi-arm studies in network meta-analysis. *Statistics in Medicine*, **33**, 4353–69

Rücker G, Schwarzer G (2015): Ranking treatments in frequentist network meta-analysis works without resampling methods. *BMC Medical Research Methodology*, **15**, 58

Rücker G, Schwarzer G (2016): Automated drawing of network plots in network meta-analysis. *Research Synthesis Methods*, **7**, 94–107

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

Rücker G, Petropoulou M, Schwarzer G (2018): Network meta-analysis of multicomponent interventions. *Manuscript submitted for publication*.

Schwarzer G, Carpenter JR and Rücker G (2015): *Meta-Analysis with R (Use-R!)*. Springer International Publishing, Switzerland. http://www.springer.com/gp/book/9783319214153

`as.data.frame.netmeta` *Additional functions for objects of class netmeta*

Description

The `as.data.frame` method returns a data frame containing information on individual studies, e.g., estimated treatment effect and its standard error.

Usage

```
## S3 method for class 'netmeta'
as.data.frame(x, row.names=NULL, optional=FALSE, details=FALSE, ...)
```

Arguments

`x`	An object of class `netmeta`.
`row.names`	`NULL` or a character vector giving the row names for the data frame.
`optional`	A logical. If `TRUE`, setting row names and converting column names (to syntactic names) is optional.
`details`	A logical. If `TRUE`, additional variables of less interest are included in data frame.
`...`	Additional arguments.

Value

A data frame is returned by the function `as.data.frame`.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

`netmeta`

Examples

```
data(Senn2013)

#
# Conduct network meta-analysis
#
net1 <- netmeta(TE, seTE, treat1, treat2, studlab, data=Senn2013, sm="MD")

as.data.frame(net1)

as.data.frame(net1, details=TRUE)
```

`decomp.design` *Design-based decomposition of Cochran's Q in network meta-analysis*

Description

This function performs a design-based decomposition of Cochran's Q for assessing the homogeneity in the whole network, the homogeneity within designs, and the homogeneity/consistency between designs. It allows also an assessment of the consistency assumption after detaching the effect of single designs.

Usage

```
decomp.design(x, tau.preset=x$tau.preset, warn=TRUE)
```

Arguments

`x`	An object of class `netmeta`.
`tau.preset`	An optional value for the square-root of the between-study variance τ^2 (see Details).
`warn`	A logical indicating whether warnings should be printed.

Details

In the context of network meta-analysis and the assessment of the homogeneity and consistency assumption, a generalized Cochran's Q statistic for multivariate meta-analysis can be used as shown in Krahn et al. (2013). This Q statistic can be decomposed in a sum of within-design Q statistics and one between-designs Q statistic that incorporates the concept of design inconsistency, see Higgins et al. (2012).

For assessing the inconsistency in a random effects model, the between-designs Q statistic can be calculated based on a full design-by-treatment interaction random effects model (see Higgins et al., 2012). This Q statistic will be automatically given in the output (tau^2 estimated by the method of moments (see Jackson et al., 2012). Alternatively, the square-root of the between-study variance can be prespecified by argument `tau.preset` to obtain a between-designs Q statistic (in `Q.inc.random`), its design-specific contributions `Q.inc.design.random.preset`) as well as residuals after detaching of single designs (`residuals.inc.detach.random.preset`).

Since an inconsistent treatment effect of one design can simultaneously inflate several residuals, Krahn et al. (2013) suggest for locating the inconsistency in a network to fit a set of extended models allowing for example for a deviating effect of each study design in turn. The recalculated between-designs Q statistics are given in list component `Q.inc.detach`. The change of the inconsistency contribution of single designs can be investigated in more detail by a net heat plot (see function netheat). Designs where only one treatment is involved in other designs of the network or where the removal of corresponding studies would lead to a splitting of the network do not contribute to the inconsistency assessment. These designs are not included in `Q.inc.detach`.

Value

A list containing the following components:

`Q.decomp` Data frame with Q statistics (variable `Q`) based on the fixed effects model to assess the homogeneity/consistency in the whole network, within designs, and between designs. Corresponding degrees of freedom (`df`) and p-values (`p.val`) are also given.

`Q.het.design` Data frame with design-specific decomposition of the within-designs Q statistic (`Q`) of the fixed effects model, corresponding degrees of freedom (`df`) and p-values (`p.val`) are given.

`Q.inc.detach` Data frame with between-designs Q statistics (`Q`) of the fixed effects model after detaching of single designs, corresponding degrees of freedom (`df`) and p-values (`p.val`) are given.

`Q.inc.design` A named vector with contributions of single designs to the between design Q statistic given in `Q.decomp`.

`Q.inc.random` Data frame with between-designs Q statistic (`Q`) based on a random effects model with square-root of between-study variance `tau.within` estimated embedded in a full design-by-treatment interaction model, corresponding degrees of freedom (`df`) and p-value (`p.val`).

`Q.inc.random.preset`
Data frame with between-designs Q statistic (`Q`) based on a random effects model with prespecified square-root of between-study variance `tau.preset` in the case if argument `tau.preset` is not NULL, corresponding degrees of freedom (`df`) and p-value (`p.val`).

`Q.inc.design.random.preset`
A named vector with contributions of single designs to the between design Q statistic based on a random effects model with prespecified square-root of between-study variance `tau.preset` in the case if argument `tau.preset` is given.

`residuals.inc.detach`
Matrix with residuals, i.e. design-specific direct estimates minus the corresponding network estimates after detaching the design of the column.

`residuals.inc.detach.random.preset`
Matrix with residuals analogous to `residuals.inc.detach` but based on a random effects model with prespecified square-root of between-study variance `tau.preset` in the case if argument `tau.preset` is not NULL.

`call` Function call.

`version` Version of R package netmeta used to create object.

Author(s)

Ulrike Krahn <ulrike.krahn@bayer.com>, Jochem König <koenigjo@uni-mainz.de>

References

Higgins JPT, Jackson D, Barrett JK, Lu G, Ades AE, White IR (2012): Consistency and inconsistency in network meta-analysis: concepts and models for multi-arm studies. *Research Synthesis Methods*, **3**, 98–110

Krahn U, Binder H, König J (2013): A graphical tool for locating inconsistency in network meta-analyses. *BMC Medical Research Methodology*, **13**, 35

Jackson D, White IR and Riley RD (2012): Quantifying the impact of between-study heterogeneity in multivariate meta-analyses. *Statistics in Medicine*, **31**, 3805–20

See Also

netmeta, netheat

Examples

```
data(Senn2013)

#
# Generation of an object of class 'netmeta' with
# reference treatment 'plac', i.e. placebo
#
net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD", reference="plac")

#
# Decomposition of Cochran's Q
#
decomp.design(net1)
```

---

dietaryfat *Network meta-analysis of dietary fat*

Description

Network meta-analysis comparing the effects of two diets to control on mortality.

The data are rates, given as the number of deaths and person-years. These data are used as an example in the supplemental material of Dias et al. (2013).

Usage

```
data(dietaryfat)
```

Format

A data frame with the following columns:

treat1 Treatment 1

treat2 Treatment 2

treat3 Treatment 3

years1 Person years arm 1

years2 Person years arm 2

years3 Person years arm 3

d1 events (deaths) arm 1

d2 events (deaths) arm 2

d3 events (deaths) arm 3

ID Study ID

Source

Dias S, Sutton AJ, Ades AE and Welton NJ (2013): Evidence synthesis for decision making 2: A generalized linear modeling framework for pairwise and network meta-analysis of randomized controlled trials. *Medical Decision Making* **33**, 607–17

See Also

pairwise, metainc, netmeta, netgraph

Examples

```
data(dietaryfat)

# Transform data from arm-based format to contrast-based format
# Using incidence rate ratios (sm="IRR") as effect measure.
# Note, the argument 'sm' is not necessary as this is the default
# in R function metainc called internally
p1 <- pairwise(list(treat1, treat2, treat3),
               list(d1, d2, d3),
               time=list(years1, years2, years3),
               studlab=ID,
               data=dietaryfat, sm="IRR")
p1

# Conduct network meta-analysis:
net1 <- netmeta(p1)
summary(net1)

# Conduct network meta-analysis using incidence rate differences
# (sm="IRD").
p2 <- pairwise(list(treat1, treat2, treat3),
               list(d1, d2, d3),
               time=list(years1, years2, years3),
```

```
              studlab=ID,
              data=dietaryfat, sm="IRD")
net2 <- netmeta(p2)
summary(net2)

# Draw network graph
netgraph(net1, points=TRUE, cex.points=3, cex=1.25)
tname <- c("Control","Diet", "Diet 2")
netgraph(net1, points=TRUE, cex.points=3, cex=1.25, labels=tname)
```

`discomb` *Additive network meta-analysis for combinations of treatments (disconnected networks)*

Description

Some treatments in a network meta-analysis may be combinations of other treatments or have common components. The influence of individual components can be evaluated in an additive network meta-analysis model assuming that the effect of treatment combinations is the sum of the effects of its components. This function implements this additive model in a frequentist way and is particularly intended for disconnected networks.

Usage

```
discomb(TE, seTE,
        treat1, treat2,
        studlab, data = NULL, subset = NULL,
        ##
        inactive = NULL,
        sep.comps = "+",
        C.matrix,
        ##
        sm,
        level = gs("level"),
        level.comb = gs("level.comb"),
        comb.fixed = gs("comb.fixed"),
        comb.random = gs("comb.random") | !is.null(tau.preset),
        ##
        reference.group = "",
        baseline.reference = TRUE,
        seq = NULL,
        ##
        tau.preset = NULL,
        ##
        tol.multiarm = 0.0005,
        details.chkmultiarm = FALSE,
        ##
        sep.trts = ":",
```

```
          nchar.trts = 666,
          ##
          backtransf = gs("backtransf"),
          ##
          title = "",
          warn = TRUE)
```

Arguments

TE	Estimate of treatment effect, i.e. difference between first and second treatment (e.g. log odds ratio, mean difference, or log hazard ratio).
seTE	Standard error of treatment estimate.
treat1	Label/Number for first treatment.
treat2	Label/Number for second treatment.
studlab	An optional - but important! - vector with study labels (see netmeta).
data	An optional data frame containing the study information.
subset	An optional vector specifying a subset of studies to be used.
inactive	A character string defining the inactive treatment (see Details).
sep.comps	A single character to define separator between treatment components.
C.matrix	C matrix (see Details).
sm	A character string indicating underlying summary measure, e.g., "RD", "RR", "OR", "ASD", "HR", "MD", "SMD", or "ROM".
level	The level used to calculate confidence intervals for individual comparisons.
level.comb	The level used to calculate confidence intervals for pooled estimates.
comb.fixed	A logical indicating whether a fixed effects (common effects) network meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects network meta-analysis should be conducted.
reference.group	Reference treatment.
baseline.reference	A logical indicating whether results should be expressed as comparisons of other treatments versus the reference treatment (default) or vice versa. This argument is only considered if reference.group has been specified.
seq	A character or numerical vector specifying the sequence of treatments in print-outs.
tau.preset	An optional value for the square-root of the between-study variance τ^2.
tol.multiarm	A numeric for the tolerance for consistency of treatment estimates and corresponding variances in multi-arm studies which are consistent by design.
details.chkmultiarm	A logical indicating whether treatment estimates and / or variances of multi-arm studies with inconsistent results or negative multi-arm variances should be printed.

sep.trts	A character used in comparison names as separator between treatment labels.
backtransf	A logical indicating whether results should be back transformed in printouts and forest plots. If backtransf=TRUE, results for sm="OR" are presented as odds ratios rather than log odds ratios, for example.
nchar.trts	A numeric defining the minimum number of characters used to create unique treatment names (see Details).
title	Title of meta-analysis / systematic review.
warn	A logical indicating whether warnings should be printed (e.g., if studies are excluded from meta-analysis due to zero standard errors).

Details

Treatments in network meta-analysis (NMA) can be complex interventions. Some treatments may be combinations of others or have common components. The standard analysis provided by netmeta is a NMA where all existing (single or combined) treatments are considered as different nodes in the network. Exploiting the fact that some treatments are combinations of common components, an additive component network meta-analysis (CNMA) model can be used to evaluate the influence of individual components. This model assumes that the effect of a treatment combination is the sum of the effects of its components which implies that common components cancel out in comparisons.

This R function can be used for disconnected networks. Use netmeta and netcomb for connected networks.

The additive CNMA model has been implemented using Bayesian methods (Mills et al., 2012; Welton et al., 2013). This function implements the additive model in a frequentist way (Rücker et al., 2018).

The underlying multivariate model is given by

$$\boldsymbol{\delta} = \boldsymbol{B\theta}, \boldsymbol{\theta} = \boldsymbol{C\beta}$$

with

- $\boldsymbol{\delta}$ vector of true treatment effects (differences) from individual studies,
- $\boldsymbol{B}$ is a design matrix describing the structure of the network,
- $\boldsymbol{\theta}$ parameter vector that represents the existing combined treatments,
- $\boldsymbol{C}$ matrix describing how the treatments are composed,
- $\boldsymbol{\beta}$ is a parameter vector representing the treatment components.

All parameters are estimated using weighted least squares regression.

Argument inactive can be used to specify a single component that does not have any therapeutic value. Accordingly, it is assumed that the treatment effect of the combination of this component with an additional treatment component is equal to the treatment effect of the additional component alone.

Argument sep.comps can be used to specify the separator between individual components. By default, the matrix **C** is calculated internally from treatment names. However, it is possible to specify a different matrix using argument C.matrix.

Value

An object of classes discomb and netcomb with corresponding print, summary, and forest functions. The object is a list containing the following components:

inactive, sep.comps, C.matrix, sm
: As defined above.

level.comb, comb.fixed, comb.random
: As defined above.

reference.group, baseline.reference, seq
: As defined above.

tau.preset, sep.trts, nchar.trts, backtransf
: As defined above.

k
: Total number of studies.

m
: Total number of pairwise comparisons.

n
: Total number of treatments.

c
: Total number of components.

comparisons.fixed, comparisons.random
: Lists with components studlab, treat1, treat2, TE, seTE, lower, upper, z, p level, df (fixed and random effects model).

components.fixed, components.random
: Lists with components TE, seTE, lower, upper, z, p level, df (fixed and random effects model).

combinations.fixed, combinations.random
: Lists with components TE, seTE, lower, upper, z, p level, df (fixed and random effects model).

Q.additive
: Overall heterogeneity / inconsistency statistic (additive model).

df.Q.additive
: Degrees of freedom for test of heterogeneity / inconsistency (additive model).

pval.Q.additive
: P-value for test of heterogeneity / inconsistency (additive model).

Q.standard
: Overall heterogeneity / inconsistency statistic (standard model).

df.Q.standard
: Degrees of freedom for test of heterogeneity / inconsistency (standard model).

pval.Q.standard
: P-value for test of heterogeneity / inconsistency (standard model).

Q.diff
: Test statistic for difference in goodness of fit between standard and additive model.

df.Q.diff
: Degrees of freedom for difference in goodness of fit between standard and additive model.

pval.Q.diff
: P-value for difference in goodness of fit between standard and additive model.

B.matrix
: Edge-vertex incidence matrix (*m*x*n*).

X
: Full design matrix (*m*x*n*).

trts
: Treatments included in network meta-analysis.

title
: Title of meta-analysis / systematic review.

call
: Function call.

version
: Version of R package netmeta used to create object.

Author(s)

Gerta Rücker <ruecker@imbi.uni-freiburg.de>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

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

Mills EJ, Thorlund K, Ioannidis JP (2012): Calculating additive treatment effects from multiple randomized trials provides useful estimates of combination therapies. *Journal of Clinical Epidemiology*, **65**, 1282–8

Rücker G, Petropoulou M, Schwarzer G (2018): Network meta-analysis of multicomponent interventions. *Manuscript submitted for publication*.

Welton NJ, Caldwell DM, Adamopoulos E, Vedhara K (2009): Mixed treatment comparison meta-analysis of complex interventions: psychological interventions in coronary heart disease. *American Journal of Epidemiology*, **169**: 1158–65

See Also

netcomb, forest.netcomb, summary.netcomb, netmeta, netconnection

Examples

```
# Artificial dataset
#
t1 <- c("A + B", "A + C", "A"    , "A"    , "D", "D", "E")
t2 <- c("C"    , "B"    , "B + C", "A + D", "E", "F", "F")
#
mean    <- c(4.1, 2.05, 0, 0, 0.1, 0.1, 0.05)
se.mean <- rep(0.1, 7)
##
study <- paste("study", c(1:4, 5, 5, 5))
#
dat <- data.frame(mean, se.mean, t1, t2, study,
                  stringsAsFactors = FALSE)
#
trts <- c("A", "A + B", "A + C", "A + D",
          "B", "B + C", "C", "D", "E", "F")
#
comps <- LETTERS[1:6]


# Use netconnection() to display network information
#
netconnection(t1, t2, study)


dc1 <- discomb(mean, se.mean, t1, t2, study, seq = trts)
dc1

forest(dc1, ref = "F")
```

```
# Define C matrix manually (which will produce the same results)
#
C <- rbind(c(1, 0, 0, 0, 0, 0),  # A
           c(1, 1, 0, 0, 0, 0),  # A + B
           c(1, 0, 1, 0, 0, 0),  # A + C
           c(1, 0, 0, 1, 0, 0),  # A + D
           c(0, 1, 0, 0, 0, 0),  # B
           c(0, 1, 1, 0, 0, 0),  # B + C
           c(0, 0, 1, 0, 0, 0),  # C
           c(0, 0, 0, 1, 0, 0),  # D
           c(0, 0, 0, 0, 1, 0),  # E
           c(0, 0, 0, 0, 0, 1))  # F
#
colnames(C) <- comps
rownames(C) <- trts
#
dc2 <- discomb(mean, se.mean, t1, t2, study, seq = trts,
               C.matrix = C)
#
# Compare C matrices
#
all.equal(dc1$C.matrix, dc2$C.matrix)
```

Dong2013 *Network meta-analysis for chronic obstructive pulmonary disease*

Description

Network meta-analysis comparing inhaled medications in patients with chronic obstructive pulmonary disease.

Usage

```
data(Dong2013)
```

Format

A data frame with the following columns:

id Study ID

treatment Treatment

death Mortality

randomized Number of individuals in treatment arm

Source

Dong Y-H, Lin H-H, Shau W-Y, Wu Y-C, Chang C-H, Lai M-S (2013): Comparative safety of inhaled medications in patients with chronic obstructive pulmonary disease: systematic review and mixed treatment comparison meta-analysis of randomised controlled trials. *Thorax*, **68**, 48–56

See Also

pairwise, metabin, netmetabin

Examples

```
data(Dong2013)

# Only consider first ten studies (to reduce runtime of example)
#
first10 <- subset(Dong2013, id <= 10)

# Transform data from long arm-based format to contrast-based format.
# Argument 'sm' has to be used for odds ratio as summary measure; by
# default the risk ratio is used in the metabin function called
# internally.
p1 <- pairwise(treatment, death, randomized, studlab = id,
               data = first10, sm = "OR")

# Conduct Mantel-Haenszel network meta-analysis
netmetabin(p1, ref = "plac")

## Not run:
# Conduct Mantel-Haenszel network meta-analysis for the whole dataset
#
p2 <- pairwise(treatment, death, randomized, studlab = id,
               data = Dong2013, sm = "OR")
netmetabin(p2, ref = "plac")

## End(Not run)
```

---

forest.netbind *Forest plot showing results of two or more network meta-analyses*

Description

Forest plot to show network estimates of two or more network meta-analyses.

Usage

```
## S3 method for class 'netbind'
forest(x,
       pooled = ifelse(x$comb.random, "random", "fixed"),
       equal.size = FALSE,
```

```
leftcols = "studlab", leftlabs = "Treatment",
rightcols = c("effect", "ci"), rightlabs = NULL,
digits = gs("digits.forest"),
digits.prop = max(gs("digits.pval") - 2, 2),
backtransf = x$backtransf,
lab.NA = "", smlab, ...)
```

Arguments

x	An object of class netbind.
pooled	A character string indicating whether results for the fixed effects ("fixed") or random effects model ("random") should be plotted. Can be abbreviated.
equal.size	A logical indicating whether all squares should be of equal size. Otherwise, the square size is proportional to the precision of estimates.
leftcols	A character vector specifying columns to be plotted on the left side of the forest plot (see Details).
leftlabs	A character vector specifying labels for columns on left side of the forest plot.
rightcols	A character vector specifying columns to be plotted on the right side of the forest plot (see Details).
rightlabs	A character vector specifying labels for columns on right side of the forest plot.
digits	Minimal number of significant digits for treatment effects and confidence intervals, see print.default.
digits.prop	Minimal number of significant digits for the direct evidence proportion.
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.
lab.NA	A character string to label missing values.
smlab	A label printed at top of figure. By default, text indicating either fixed effect or random effects model is printed.
...	Additional arguments for forest.meta function.

Details

A forest plot, also called confidence interval plot, is drawn in the active graphics window.

The arguments leftcols and rightcols can be used to specify columns which are plotted on the left and right side of the forest plot, respectively. If argument rightcols is FALSE, no columns will be plotted on the right side.

For more information see help page of forest.meta function.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

netbind, netcomb, forest.meta

Examples

```
data(Linde2016)

# Only consider studies including Face-to-face PST (to reduce runtime
# of example)
#
face <- subset(Linde2016, id %in% c(16, 24, 49, 118))

# Standard random effects NMA model (with placebo as reference treatment)
#
net1 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = face, reference.group = "placebo",
                sm = "OR", comb.fixed = FALSE)

# Additive CNMA model with placebo as inactive component and reference
#
nc1 <- netcomb(net1, inactive = "placebo")

# Combine results of standard NMA and CNMA
#
nb1 <- netbind(nc1, net1,
               name = c("Additive CNMA", "Standard NMA"),
               col.study = c("red", "black"),
               col.square = c("red", "black"))
forest(nb1,
       col.by = "black", addrow.subgroups = FALSE,
       fontsize = 10, spacing = 0.7, squaresize = 0.9,
       label.left = "Favours Placebo",
       label.right = "Favours other")
```

---

forest.netcomb *Forest plot for additive network meta-analysis*

Description

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

Usage

```
## S3 method for class 'netcomb'
forest(x,
       pooled=ifelse(x$comb.random, "random", "fixed"),
       reference.group=x$reference.group,
       baseline.reference=x$baseline.reference,
       leftcols="studlab", leftlabs="Treatment",
       rightcols=c("effect", "ci"), rightlabs=NULL,
       digits=gs("digits.forest"), small.values="good",
       smlab=NULL,
       sortvar=x$seq, backtransf=x$backtransf,
```

```
lab.NA=".", add.data,
drop.reference.group = FALSE,
...)
```

Arguments

x	An object of class netcomb.
reference.group	Reference treatment(s).
baseline.reference	A logical indicating whether results should be expressed as comparisons of other treatments versus the reference treatment (default) or vice versa.
pooled	A character string indicating whether results for the fixed effect ("fixed") or random effects model ("random") should be plotted. Can be abbreviated.
leftcols	A character vector specifying (additional) columns to be plotted on the left side of the forest plot or a logical value (see forest.meta help page for details).
leftlabs	A character vector specifying labels for (additional) columns on left side of the forest plot (see forest.meta help page for details).
rightcols	A character vector specifying (additional) columns to be plotted on the right side of the forest plot or a logical value (see forest.meta help page for details).
rightlabs	A character vector specifying labels for (additional) columns on right side of the forest plot (see forest.meta help page for details).
digits	Minimal number of significant digits for treatment effects and confidence intervals, see print.default.
small.values	A character string specifying whether small treatment effects indicate a beneficial ("good") or harmful ("bad") effect, can be abbreviated; see netrank.
smlab	A label printed at top of figure. By default, text indicating either fixed effect or random effects model is printed.
sortvar	An optional vector used to sort the individual studies (must be of same length as the total number of treatments).
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.
lab.NA	A character string to label missing values.
add.data	An optional data frame with additional columns to print in forest plot (see Details).
drop.reference.group	A logical indicating whether the reference group should be printed in the forest plot.
...	Additional arguments for forest.meta function.

Details

A forest plot, also called confidence interval plot, is drawn in the active graphics window.

Argument `sortvar` can be either a numeric or character vector with length of number of treatments. If `sortvar` is numeric the `order` function is utilised internally to determine the order of values. If `sortvar` is character it must be a permutation of the treatment names. It is also possible to to sort by treatment comparisons (`sortvar=TE`, etc.), standard error (`sortvar=seTE`), and number of studies with direct treatment comparisons (`sortvar=k`).

Argument `add.data` can be used to add additional columns to the forest plot. This argument must be a data frame with the same row names as the treatment effects matrices in R object `x`, i.e., `x$TE.fixed` or `x$TE.random`.

For more information see help page of `forest.meta` function.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

`netcomb`, `discomb`, `forest.meta`

Examples

```
data(Linde2016)

# Only consider studies including Face-to-face PST (to reduce runtime
# of example)
#
face <- subset(Linde2016, id %in% c(16, 24, 49, 118))

# Conduct random effects network meta-analysis
#
net1 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = face, ref = "placebo",
                sm = "OR", comb.fixed = FALSE)

# Additive model for treatment components (with placebo as inactive
# treatment)
#
nc1 <- netcomb(net1, inactive = "placebo")
#
forest(nc1)

## Not run:
# Specify, order of treatments
#
trts <- c("TCA", "SSRI", "SNRI", "NRI", "Low-dose SARI", "NaSSa",
          "rMAO-A", "Ind drug", "Hypericum", "Face-to-face CBT",
          "Face-to-face PST", "Face-to-face interpsy", "Face-to-face psychodyn",
          "Other face-to-face", "Remote CBT", "Self-help CBT", "No contact CBT",
          "Face-to-face CBT + SSRI", "Face-to-face interpsy + SSRI",
```

```
          "Face-to-face PST + SSRI", "UC", "Placebo")
#
# Note, three treatments are actually combinations of 'SSRI' with
# other components:
#
# "Face-to-face CBT + SSRI",
# "Face-to-face interpsy + SSRI",
# "Face-to-face PST + SSRI"

# Conduct random effects network meta-analysis
#
net2 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = Linde2016, ref = "placebo",
                seq = trts,
                sm = "OR", comb.fixed = FALSE)
#
summary(net2)

# Additive model for treatment components (with placebo as inactive
# treatment)
#
nc2 <- netcomb(net2, inactive = "placebo")
#
forest(nc2)

## End(Not run)
```

`forest.netmeta` *Forest plot for network meta-analysis*

Description

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

Usage

```
## S3 method for class 'netmeta'
forest(x,
       pooled=ifelse(x$comb.random, "random", "fixed"),
       reference.group=x$reference.group,
       baseline.reference=x$baseline.reference,
       leftcols="studlab", leftlabs,
       rightcols=c("effect", "ci"), rightlabs,
       digits=gs("digits.forest"), small.values="good", digits.Pscore=2,
       smlab=NULL,
       sortvar=x$seq, backtransf=x$backtransf,
       lab.NA=".", add.data,
       drop.reference.group = FALSE,
       col.by = "black", print.byvar = FALSE,
       ...)
```

Arguments

x	An object of class netmeta.
reference.group	Reference treatment(s).
baseline.reference	A logical indicating whether results should be expressed as comparisons of other treatments versus the reference treatment (default) or vice versa.
pooled	A character string indicating whether results for the fixed effect ("fixed") or random effects model ("random") should be plotted. Can be abbreviated.
leftcols	A character vector specifying (additional) columns to be plotted on the left side of the forest plot or a logical value (see forest.meta help page for details).
leftlabs	A character vector specifying labels for (additional) columns on left side of the forest plot (see forest.meta help page for details).
rightcols	A character vector specifying (additional) columns to be plotted on the right side of the forest plot or a logical value (see forest.meta help page for details).
rightlabs	A character vector specifying labels for (additional) columns on right side of the forest plot (see forest.meta help page for details).
digits	Minimal number of significant digits for treatment effects and confidence intervals, see print.default.
small.values	A character string specifying whether small treatment effects indicate a beneficial ("good") or harmful ("bad") effect, can be abbreviated; see netrank.
digits.Pscore	Minimal number of significant digits for P-score, see print.default and netrank.
smlab	A label printed at top of figure. By default, text indicating either fixed effect or random effects model is printed.
sortvar	An optional vector used to sort the individual studies (must be of same length as the total number of treatments).
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.
lab.NA	A character string to label missing values.
add.data	An optional data frame with additional columns to print in forest plot (see Details).
drop.reference.group	A logical indicating whether the reference group should be printed in the forest plot.
col.by	The colour to print information on subgroups.
print.byvar	A logical indicating whether the name of the grouping variable should be printed in front of the group labels.
...	Additional arguments for forest.meta function.

Details

A forest plot, also called confidence interval plot, is drawn in the active graphics window.

Argument `sortvar` can be either a numeric or character vector with length of number of treatments. If `sortvar` is numeric the `order` function is utilised internally to determine the order of values. If `sortvar` is character it must be a permutation of the treatment names. It is also possible to provide either `sortvar=Pscore`, `sortvar="Pscore"`, `sortvar=-Pscore` or `sortvar="-Pscore"` in order to sort treatments according to the ranking generated by `netrank` which is called internally. Similar expressions are possible to sort by treatment comparisons (`sortvar=TE`, etc.), standard error (`sortvar=seTE`), number of studies with direct treatment comparisons (`sortvar=k`), and direct evidence proportion (`sortvar=prop.direct`, see also `netmeasures`).

Argument `add.data` can be used to add additional columns to the forest plot. This argument must be a data frame with the same row names as the treatment effects matrices in R object `x`, i.e., `x$TE.fixed` or `x$TE.random`.

For more information see help page of `forest.meta` function.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

`forest.meta`

Examples

```
data(Senn2013)

## Not run:
#
# Conduct network meta-analysis
#
net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD")

forest(net1, ref="plac")

forest(net1, xlim=c(-1.5,1), ref="plac",
       xlab="HbA1c difference", rightcols=FALSE)

## End(Not run)

#
# Random effects effect model
#
net2 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD", comb.fixed=FALSE)

forest(net2, xlim=c(-1.5,1), ref="plac",
       xlab="HbA1c difference")
```

```
## Not run:
#
# Add column with P-Scores on right side of forest plot
#
forest(net2, xlim=c(-1.5,1), ref="plac",
       xlab="HbA1c difference",
       rightcols=c("effect", "ci", "Pscore"),
       rightlabs="P-Score",
       just.addcols="right")

#
# Add column with P-Scores on left side of forest plot
#
forest(net2, xlim=c(-1.5,1), ref="plac",
       xlab="HbA1c difference",
       leftcols=c("studlab", "Pscore"),
       leftlabs=c("Treatment", "P-Score"),
       just.addcols="right")

#
# Sort forest plot by descending P-Score
#
forest(net2, xlim=c(-1.5,1), ref="plac",
       xlab="HbA1c difference",
       rightcols=c("effect", "ci", "Pscore"),
       rightlabs="P-Score",
       just.addcols="right",
       sortvar=-Pscore)

#
# Drop reference group and sort by and print number of studies
# with direct treatment comparisons
#
forest(net2, xlim=c(-1.5,1), ref="plac",
       xlab="HbA1c difference",
       leftcols=c("studlab", "k"),
       leftlabs=c("Contrast\nto Placebo", "Direct\nComparisons"),
       sortvar=-k,
       drop=TRUE,
       smlab="Random Effects Model")

## End(Not run)
```

`forest.netsplit` *Forest plot for direct and indirect evidence*

Description

Forest plot to show direct and indirect evidence in network meta-analysis. Furthermore, estimates from network meta-analysis as well as prediction intervals can be printed.

Usage

```
## S3 method for class 'netsplit'
forest(x,
       pooled=ifelse(x$comb.random, "random", "fixed"),
       show = "both",
       subgroup = "comparison",
       overall = TRUE, direct = TRUE, indirect = TRUE,
       prediction = x$prediction,
       text.overall = "Network estimate",
       text.direct = "Direct estimate",
       text.indirect = "Indirect estimate",
       text.predict = "Prediction interval",
       type.overall, type.direct, type.indirect,
       col.square = "gray", col.square.lines = col.square,
       col.inside = "white",
       col.diamond = "gray", col.diamond.lines = "black",
       col.predict = "red", col.predict.lines = "black",
       equal.size = FALSE,
       leftcols, leftlabs,
       rightcols = c("effect", "ci"), rightlabs = NULL,
       digits = gs("digits.forest"),
       digits.prop = max(gs("digits.pval") - 2, 2),
       backtransf = x$backtransf,
       lab.NA = "", smlab, ...)
```

Arguments

x	An object of class netsplit.
pooled	A character string indicating whether results for the fixed effect ("fixed") or random effects model ("random") should be plotted. Can be abbreviated.
show	A character string indicating which comparisons should be printed (see Details).
overall	A logical indicating whether network meta-analysis estimates should be printed.
direct	A logical indicating whether direct estimates should be printed.
indirect	A logical indicating whether indirect estimates should be printed.
prediction	A logical indicating whether prediction intervals should be printed.
subgroup	A character string indicating which layout should be used in forest plot: subgroups by comparisons ("comparison") or subgroups by estimates ("estimate"). Can be abbreviated.
text.overall	A character string used in the plot to label the network estimates.
text.direct	A character string used in the plot to label the direct estimates.
text.indirect	A character string used in the plot to label the indirect estimates.
text.predict	A character string used in the plot to label the prediction interval.
type.overall	A character string specifying how to plot treatment effects and confidence intervals for the overall network evidence.

type.direct	A character string specifying how to plot treatment effects and confidence intervals for the direct evidence.
type.indirect	A character string specifying how to plot treatment effects and confidence intervals for the indirect evidence.
col.square	The colour for squares.
col.square.lines	The colour for the outer lines of squares.
col.inside	The colour for results and confidence limits if confidence limits are completely within squares squares.
col.diamond	The colour of diamonds.
col.diamond.lines	The colour of the outer lines of diamonds.
col.predict	Background colour of prediction intervals.
col.predict.lines	Colour of outer lines of prediction intervals.
equal.size	A logical indicating whether all squares should be of equal size. Otherwise, the square size is proportional to the precision of estimates.
leftcols	A character vector specifying columns to be plotted on the left side of the forest plot (see Details).
leftlabs	A character vector specifying labels for columns on left side of the forest plot.
rightcols	A character vector specifying columns to be plotted on the right side of the forest plot (see Details).
rightlabs	A character vector specifying labels for columns on right side of the forest plot.
digits	Minimal number of significant digits for treatment effects and confidence intervals, see print.default.
digits.prop	Minimal number of significant digits for the direct evidence proportion.
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.
lab.NA	A character string to label missing values.
smlab	A label printed at top of figure. By default, text indicating either fixed effect or random effects model is printed.
...	Additional arguments for forest.meta function.

Details

A forest plot, also called confidence interval plot, is drawn in the active graphics window.

The arguments leftcols and rightcols can be used to specify columns which are plotted on the left and right side of the forest plot, respectively. If argument rightcols is FALSE, no columns will be plotted on the right side.

If direct estimates are included in the forest plot (direct=TRUE, default), the following columns will be printed on the left side of the forest plot: the comparisons (column "studlab" in forest.meta), number of pairwise comparisons ("k"), and direct evidence proportion ("k").

If direct estimates are not included in the forest plot (`direct=FALSE`), only the comparisons (`"studlab"`) are printed on the left side of the forest plot.

For more information see help page of `forest.meta` function.

Argument `show` determines which comparisons are printed:

"all"	All comparisons
"both"	Only comparisons contributing both direct and indirect evidence
"with.direct"	Comparisons providing direct evidence
"direct.only"	Comparisons providing only direct evidence
"indirect.only"	Comparisons providing only indirect evidence

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

`forest.meta`

Examples

```
data(Senn2013)
#
net1 <- netmeta(TE, seTE, treat1.long, treat2.long,
                studlab, data = Senn2013,
                comb.fixed = FALSE)
#
ns1 <- netsplit(net1)

# Forest plot showing comparisons contributing both direct and
# indirect evidence
#
forest(ns1, fontsize = 6, spacing = 0.5, addrow.subgroups = FALSE)


# Forest plot showing comparisons contributing direct evidence
#
forest(ns1, fontsize = 6, spacing = 0.5, addrow.subgroups = FALSE,
       show = "with.direct")
```

`funnel.netmeta` *'Comparison-adjusted' funnel plot*

Description

Draw a 'comparison-adjusted' funnel plot to assess funnel plot asymmetry in network meta-analysis.

Usage

```
## S3 method for class 'netmeta'
funnel(x,
       order,
       pooled = ifelse(x$comb.random, "random", "fixed"),
       ##
       xlab,
       level = x$level,
       ##
       pch,
       col = "black",
       ##
       legend = TRUE,
       linreg = FALSE,
       rank = FALSE,
       mm = FALSE,
       ##
       pos.legend = "topright",
       pos.tests = "topleft",
       ##
       text.linreg = "(Egger)",
       text.rank = "(Begg-Mazumdar)",
       text.mm = "(Thompson-Sharp)",
       ##
       sep.trts = x$sep.trts,
       nchar.trts = x$nchar.trts,
       ##
       backtransf = x$backtransf,
       digits.pval = gs("digits.pval"),
       ...)
```

Arguments

x	An object of class netmeta.
order	A mandatory character or numerical vector specifying the order of treatments (see Details).
pooled	A character string indicating whether results for the fixed effect ("fixed") or random effects model ("random") should be plotted. Can be abbreviated.
xlab	A label for the x-axis.
level	The confidence level utilised in the plot. For the funnel plot, confidence limits are not drawn if yaxis="size".
pch	The plotting symbol(s) used for individual studies within direct comparisons.
col	The colour(s) used for individual studies within direct comparisons.
legend	A logical indicating whether a legend with information on direct comparisons should be added to the plot.

linreg	A logical indicating whether result of linear regression test for funnel plot asymmetry should be added to plot.
rank	A logical indicating whether result of rank test for funnel plot asymmetry should be added to plot.
mm	A logical indicating whether result of linear regression test for funnel plot asymmetry allowing for between-study heterogeneity should be added to plot.
pos.legend	The position of the legend describing plotting symbols and colours for direct comparisons.
pos.tests	The position of results for test(s) of funnel plot asymmetry.
text.linreg	A character string used in the plot to label the linear regression test for funnel plot asymmetry.
text.rank	A character string used in the plot to label the rank test for funnel plot asymmetry.
text.mm	A character string used in the plot to label the linear regression test for funnel plot asymmetry allowing for between-study heterogeneity.
sep.trts	A character used in comparison names as separator between treatment labels.
nchar.trts	A numeric defining the minimum number of characters used to create unique treatment names (see netmeta).
backtransf	A logical indicating whether results for relative summary measures (argument sm equal to "OR", "RR", "HR", or "IRR") should be back transformed in funnel plots. If backtransf=TRUE, results for sm="OR" are printed as odds ratios rather than log odds ratios, for example.
digits.pval	Minimal number of significant digits for p-value of test(s) for funnel plot asymmetry.
...	Additional graphical arguments passed as arguments to funnel.meta.

Details

A 'comparison-adjusted' funnel plot (Chaimani & Salanti, 2012) is drawn in the active graphics window.

Argument order is mandatory to determine the order of treatments (Chaimani et al., 2013):

"Before using this plot, investigators should order the treatments in a meaningful way and make assumptions about how small studies differ from large ones. For example, if they anticipate that newer treatments are favored in small trials, then they could name the treatments from oldest to newest so that all comparisons refer to 'old versus new intervention'. Other possibilities include defining the comparisons so that all refer to an active treatment versus placebo or sponsored versus non-sponsored intervention."

The treatments can be either in increasing or decreasing order.

In the funnel plot, if yaxis is "se", the standard error of the treatment estimates is plotted on the y-axis which is likely to be the best choice (Sterne & Egger, 2001). Other possible choices for yaxis are "invvar" (inverse of the variance), "invse" (inverse of the standard error), and "size" (study size).

Value

A data frame with the following columns:

studlab	Study label.
treat1	Label/Number for first treatment.
treat2	Label/Number for second treatment.
comparison	Treatment comparison.
TE	Estimate of treatment effect, e.g., log odds ratio.
TE.direct	Pooled estimate from direct evidence.
TE.adj	'Comparison-adjusted' treatment effect (TE - TE.direct).
seTE	Standard error of treatment estimate.
pch	Plotting symbol(s).
col	Colour of plotting symbol(s).

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Chaimani A, Salanti G (2012): Using network meta-analysis to evaluate the existence of small-study effects in a network of interventions. Research Synthesis Methods, **3**, 161–76

Chaimani A, Higgins JP, Mavridis D, Spyridonos P, Salanti G (2013): Graphical tools for network meta-analysis in STATA. PLOS ONE, **8**, e76654

Sterne JAC & Egger M (2001): Funnel plots for detecting bias in meta-analysis: Guidelines on choice of axis. *Journal of Clinical Epidemiology*, **54**, 1046–55

See Also

netmeta, funnel.meta, metabias

Examples

```
data(Senn2013)
net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data = Senn2013, sm = "MD")

#
# 'Comparison-adjusted' funnel plot not created as argument 'order' is
# missing
#
funnel(net1)

#
# (Non-sensical) alphabetic order of treatments with placebo as last
# treatment
#
```

```
ord <- c("a", "b", "me", "mi", "pi", "r", "si", "su", "v", "pl")
funnel(net1, order = ord)

#
# Add results for tests of funnel plot asymmetry and use different
# plotting symbols and colours
#
funnel(net1, order = ord,
       pch = rep(c(15:18, 1), 3), col = 1:3,
       linreg = TRUE, rank = TRUE, mm = TRUE, digits.pval = 2)

#
# Same results for tests of funnel plot asymmetry using reversed order
# of treatments
#
funnel(net1, order = rev(ord),
       pch = rep(c(15:18, 1), 3), col = 1:3,
       linreg = TRUE, rank = TRUE, mm = TRUE, digits.pval = 2)

#
# Calculate tests for funnel plot asymmetry
#
f1 <- funnel(net1, order = ord,
             pch = rep(c(15:18, 1), 3), col = 1:3,
             linreg = TRUE, rank = TRUE, mm = TRUE)
#
metabias(metagen(TE.adj, seTE, data = f1))
metabias(metagen(TE.adj, seTE, data = f1), method = "rank")
metabias(metagen(TE.adj, seTE, data = f1), method = "mm")
```

Gurusamy2011 *Network meta-analysis on blood loss during liver transplantation*

Description

Network meta-analysis comparing the effects of a number of interventions for decreasing blood loss and blood transfusion requirements during liver transplantation.

Usage

```
data(Gurusamy2011)
```

Format

A data frame with the following columns:

study Study information (first author, year)

treatment Treatment

death Mortality at 60 days post-transplantation

n Number of individuals in treatment arm

Source

Gurusamy KS, Pissanou T, Pikhart H, Vaughan J, Burroughs AK, Davidson BR (2011): Methods to decrease blood loss and transfusion requirements for liver transplantation. *Cochrane Database of Systematic Reviews*, CD009052

See Also

pairwise, metabin, netmetabin

Examples

```
data(Gurusamy2011)

# Only consider three studies (to reduce runtime of example)
#
studies <- c("Findlay 2001", "Garcia-Huete 1997", "Dalmau 2000")
three <- subset(Gurusamy2011, study %in% studies)

# Transform data from long arm-based format to contrast-based format.
# Argument 'sm' has to be used for odds ratio as summary measure; by
# default the risk ratio is used in the metabin function called
# internally.
p1 <- pairwise(treatment, death, n, studlab = study,
               data = three, sm = "OR")

# Conduct Mantel-Haenszel network meta-analysis
netmetabin(p1, ref = "cont")

## Not run:
p2 <- pairwise(treatment, death, n, studlab = study,
               data = Gurusamy2011, sm = "OR")

# Conduct Mantel-Haenszel network meta-analysis
netmetabin(p2, ref = "cont")

## End(Not run)
```

---

hasse *Hasse diagram*

Description

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

Usage

```
hasse(x,
      pooled=ifelse(x$comb.random, "random", "fixed"),
      newpage = TRUE)
```

Arguments

x	An object of class netposet (mandatory).
pooled	A character string indicating whether Hasse diagram show be drawn for fixed effect ("fixed") or random effects model ("random"). Can be abbreviated.
newpage	A logical value indicating whether a new figure should be printed in an existing graphics window. Otherwise, the Hasse diagram is added to the existing figure.

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 R function hasse in R package **hasseDiagram** (Krzysztof Ciomek, https://github.com/kciomek/hasseDiagram), i.e., function hasse can only be used if R package **hasseDiagram** is installed.

Author(s)

Gerta Rücker <ruecker@imbi.uni-freiburg.de>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

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

See Also

netmeta, netposet

Examples

```
# Use depression dataset
#
data(Linde2015)
#
# Define order of treatments
#
trts <- c("TCA", "SSRI", "SNRI", "NRI",
          "Low-dose SARI", "NaSSa", "rMAO-A", "Hypericum",
          "Placebo")
#
# Outcome labels
#
outcomes <- c("Early response", "Early remission")
#
# (1) Early response
#
p1 <- pairwise(treat = list(treatment1, treatment2, treatment3),
```

```
                event = list(resp1, resp2, resp3),
                n = list(n1, n2, n3),
                studlab = id, data = Linde2015, sm = "OR")
#
net1 <- netmeta(p1, comb.fixed = FALSE,
                 seq = trts, ref = "Placebo")
#
# (2) Early remission
#
p2 <- pairwise(treat = list(treatment1, treatment2, treatment3),
                event = list(remi1, remi2, remi3),
                n = list(n1, n2, n3),
                studlab = id, data = Linde2015, sm = "OR")
#
net2 <- netmeta(p2, comb.fixed = FALSE,
                 seq = trts, ref = "Placebo")
#
# Partial order of treatment rankings
#
po <- netposet(netrank(net1, small.values = "bad"),
                netrank(net2, small.values = "bad"),
                outcomes = outcomes)
#
# Hasse diagram
#
hasse(po)
```

`Linde2015` *Network meta-analysis of treatments for depression*

Description

Network meta-analysis of nine classes of antidepressants including placebo for the primary care setting; partly shown in Linde et al. (2015), supplementary Table 2.

Usage

```
data(Linde2015)
```

Format

A data frame with the following columns:

id Study ID

author First author

year Publication year

treatment1 First treatment

treatment2 Second treatment

treatment3 Third treatment

n1 Number of patients receiving first treatment

resp1 Number of early responder (treatment 1)

remi1 Number of early remissions (treatment 1)

loss1 Number of patients loss to follow-up (treatment 1)

loss.ae1 Number of patients loss to follow-up due to adverse events (treatment 1)

ae1 Number of patients with adverse events (treatment 1)

n2 Number of patients receiving second treatment

resp2 Number of early responder (treatment 2)

remi2 Number of early remissions (treatment 2)

loss2 Number of patients loss to follow-up (treatment 2)

loss.ae2 Number of patients loss to follow-up due to adverse events (treatment 2)

ae2 Number of patients with adverse events (treatment 2)

n3 Number of patients receiving third treatment

resp3 Number of early responder (treatment 3)

remi3 Number of early remissions (treatment 3)

loss3 Number of patients loss to follow-up (treatment 3)

loss.ae3 Number of patients loss to follow-up due to adverse events (treatment 3)

ae3 Number of patients with adverse events (treatment 3)

Source

Linde K, Kriston L, Rücker G, et al. (2015): Efficacy and acceptability of pharmacological treatments for depressive disorders in primary care: Systematic review and network meta-analysis. *Annals of Family Medicine* **13**, 69–79

See Also

pairwise, metabin, netmeta, netposet

Examples

```
data(Linde2015)

# Transform data from arm-based format to contrast-based format
# Outcome: early response
p1 <- pairwise(list(treatment1, treatment2, treatment3),
               event = list(resp1, resp2, resp3),
       n = list(n1, n2, n3),
               studlab = id, data = Linde2015, sm = "OR")

# Define order of treatments
trts <- c("TCA", "SSRI", "SNRI", "NRI",
          "Low-dose SARI", "NaSSa", "rMAO-A", "Hypericum",
          "Placebo")
```

```
# Conduct random effects network meta-analysis
net1 <- netmeta(p1, comb.fixed = FALSE,
                reference = "Placebo",
seq = trts)
print(summary(net1), digits = 2)
```

Linde2016 *Network meta-analysis of primary care depression treatments*

Description

Network meta-analysis of 22 treatments (including placebo and usual care) for the primary care of depression.

Usage

```
data(Linde2016)
```

Format

A data frame with the following columns:

id Study ID

lnOR Response after treatment (log odds ratio)

selnOR Standard error of log odds ratio

treat1 First treatment

treat2 Second treatment

Source

Linde K, Rücker G, Schneider A et al. (2016): Questionable assumptions hampered interpretation of a network meta-analysis of primary care depression treatments. *Journal of Clinical Epidemiology* **71**, 86–96

See Also

netmeta, netcomb

Examples

```
data(Linde2016)

# Only consider studies including Face-to-face PST (to reduce runtime
# of example)
#
face <- subset(Linde2016, id %in% c(16, 24, 49, 118))
```

```
# Conduct random effects network meta-analysis
#
net1 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = face, reference.group = "placebo",
                sm = "OR", comb.fixed = FALSE,
                nchar = 6)
#
summary(net1)

## Not run:
# Conduct random effects network meta-analysis
#
net2 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = Linde2016, reference.group = "placebo",
                sm = "OR", comb.fixed = FALSE,
                nchar = 6)
#
summary(net2)

## End(Not run)
```

`netbind` *Combine network meta-analysis objects*

Description

This function can be used to combine network meta-analysis objects which is especially useful to generate a forest plot with results of several network meta-analyses.

Usage

```
netbind(..., name, comb.fixed, comb.random,
        col.study = "black", col.inside = "white",
        col.square = "gray", col.square.lines = col.square,
        backtransf, reference.group, baseline.reference)
```

Arguments

`...`	Any number of meta-analysis objects (see Details).
`name`	An optional character vector providing descriptive names for the network meta-analysis objects.
`comb.fixed`	A logical indicating whether results for the fixed effects (common effects) model should be reported.
`comb.random`	A logical indicating whether results for the random effects model should be reported.
`col.study`	The colour for network estimates and confidence limits.
`col.inside`	The colour for network estimates and confidence limits if confidence limits are completely within squares.

col.square The colour for squares.

col.square.lines
The colour for the outer lines of squares.

backtransf A logical indicating whether results should be back transformed. If backtransf=TRUE (default), results for sm="OR" are printed as odds ratios rather than log odds ratios, for example.

reference.group
Reference treatment.

baseline.reference
A logical indicating whether results should be expressed as comparisons of other treatments versus the reference treatment (default) or vice versa. This argument is only considered if reference.group has been specified.

Details

This function can be used to combine network meta-analysis objects which is especially useful to generate a forest plot with results of several network meta-analyses.

Value

An object of class "netbind" with corresponding forest function. The object is a list containing the following components:

fixed A data frame with results for the fixed effects model.

random A data frame with results for the random effects model.

sm Summary measure used in network meta-analyses.

level.comb Level for confidence intervals.

comb.fixed, comb.random, backtransf
As defined above.

reference.group, baseline.reference
As defined above.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

netmeta, netcomb, discomb, forest.netbind

Examples

```
data(Linde2016)

# Only consider studies including Face-to-face PST (to reduce runtime
# of example)
#
face <- subset(Linde2016, id %in% c(16, 24, 49, 118))
```

```
# Standard random effects NMA model (with placebo as reference treatment)
#
net1 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = face, reference.group = "placebo",
                sm = "OR", comb.fixed = FALSE)

# Additive CNMA model with placebo as inactive component and reference
#
nc1 <- netcomb(net1, inactive = "placebo")

# Combine results of standard NMA and CNMA
#
nb1 <- netbind(nc1, net1,
               name = c("Additive CNMA", "Standard NMA"),
               col.study = c("red", "black"),
               col.square = c("red", "black"))
forest(nb1,
       col.by = "black", addrow.subgroups = FALSE,
       fontsize = 10, spacing = 0.7, squaresize = 0.9,
       label.left = "Favours Placebo",
       label.right = "Favours other")
```

`netcomb` *Additive network meta-analysis for combinations of treatments*

Description

Some treatments in a network meta-analysis may be combinations of other treatments or have common components. The influence of individual components can be evaluated in an additive network meta-analysis model assuming that the effect of treatment combinations is the sum of the effects of its components. This function implements this additive model in a frequentist way.

Usage

```
netcomb(x, inactive = NULL, sep.comps = "+", C.matrix,
        comb.fixed = x$comb.fixed,
        comb.random = x$comb.random | !is.null(tau.preset),
        tau.preset = NULL)
```

Arguments

`x`	An object of class `netmeta`.
`inactive`	A character string defining the inactive treatment (see Details).
`sep.comps`	A single character to define separator between treatment components.
`C.matrix`	C matrix (see Details).
`comb.fixed`	A logical indicating whether a fixed effects (common effects) network meta-analysis should be conducted.

`comb.random` A logical indicating whether a random effects network meta-analysis should be conducted.

`tau.preset` An optional value for the square-root of the between-study variance τ^2.

Details

Treatments in network meta-analysis (NMA) can be complex interventions. Some treatments may be combinations of others or have common components. The standard analysis provided by `netmeta` is a NMA where all existing (single or combined) treatments are considered as different nodes in the network. Exploiting the fact that some treatments are combinations of common components, an additive component network meta-analysis (CNMA) model can be used to evaluate the influence of individual components. This model assumes that the effect of a treatment combination is the sum of the effects of its components which implies that common components cancel out in comparisons.

The additive CNMA model has been implemented using Bayesian methods (Mills et al., 2012; Welton et al., 2013). This function implements the additive model in a frequentist way (Rücker et al., 2018).

The underlying multivariate model is given by

$$\boldsymbol{\delta} = \boldsymbol{B\theta}, \boldsymbol{\theta} = \boldsymbol{C\beta}$$

with

- $\boldsymbol{\delta}$ vector of true treatment effects (differences) from individual studies,
- $\boldsymbol{B}$ is a design matrix describing the structure of the network,
- $\boldsymbol{\theta}$ parameter vector that represents the existing combined treatments,
- $\boldsymbol{C}$ matrix describing how the treatments are composed,
- $\boldsymbol{\beta}$ is a parameter vector representing the treatment components.

All parameters are estimated using weighted least squares regression.

Argument `inactive` can be used to specify a single component that does not have any therapeutic value. Accordingly, it is assumed that the treatment effect of the combination of this component with an additional treatment component is equal to the treatment effect of the additional component alone.

Argument `sep.comps` can be used to specify the separator between individual components. By default, the matrix **C** is calculated internally from treatment names. However, it is possible to specify a different matrix using argument `C.matrix`.

Value

An object of class `netcomb` with corresponding `print`, `summary`, and `forest` functions. The object is a list containing the following components:

`x, inactive, sep.comps, C.matrix`
: As defined above.

`comb.fixed, comb.random, tau.preset`
: As defined above.

`k` Total number of studies.

`m` Total number of pairwise comparisons.

`n` Total number of treatments.

`c` Total number of components.

`comparisons.fixed, comparisons.random`
Lists with components studlab, treat1, treat2, TE, seTE, lower, upper, z, p level, df (fixed and random effects model).

`components.fixed, components.random`
Lists with components TE, seTE, lower, upper, z, p level, df (fixed and random effects model).

`combinations.fixed, combinations.random`
Lists with components TE, seTE, lower, upper, z, p level, df (fixed and random effects model).

`sm` Summary measure.

`level.comb` Level for confidence intervals.

`Q.additive` Overall heterogeneity / inconsistency statistic (additive model).

`df.Q.additive` Degrees of freedom for test of heterogeneity / inconsistency (additive model).

`pval.Q.additive`
P-value for test of heterogeneity / inconsistency (additive model).

`Q.standard` Overall heterogeneity / inconsistency statistic (standard model).

`df.Q.standard` Degrees of freedom for test of heterogeneity / inconsistency (standard model).

`pval.Q.standard`
P-value for test of heterogeneity / inconsistency (standard model).

`Q.diff.fixed, Q.diff.random`
Test statistic for difference in goodness of fit between standard and additive model (fixed and random effects model).

`df.Q.diff.fixed, df.Q.diff.random`
Degrees of freedom for difference in goodness of fit between standard and additive model (fixed and random effects model).

`pval.Q.diff.fixed, pval.Q.diff.random`
P-value for difference in goodness of fit between standard and additive model (fixed and random effects model).

`backtransf` A logical indicating whether results should be back transformed in printouts and forest plots.

`nchar.trts` A numeric defining the minimum number of characters used to create unique treatment and component names.

`title` Title of meta-analysis / systematic review.

`call` Function call.

`version` Version of R package netmeta used to create object.

Author(s)

Gerta Rücker <ruecker@imbi.uni-freiburg.de>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

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

Mills EJ, Thorlund K, Ioannidis JP (2012): Calculating additive treatment effects from multiple randomized trials provides useful estimates of combination therapies. *Journal of Clinical Epidemiology*, **65**, 1282–8

Rücker G, Petropoulou M, Schwarzer G (2018): Network meta-analysis of multicomponent interventions. *Manuscript submitted for publication*.

Welton NJ, Caldwell DM, Adamopoulos E, Vedhara K (2009): Mixed treatment comparison meta-analysis of complex interventions: psychological interventions in coronary heart disease. *American Journal of Epidemiology*, **169**: 1158–65

See Also

netmeta, forest.netcomb, print.netcomb, discomb

Examples

```
data(Linde2016)

# Only consider studies including Face-to-face PST (to reduce runtime
# of example)
#
face <- subset(Linde2016, id %in% c(16, 24, 49, 118))

# Conduct random effects network meta-analysis
#
net1 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = face, ref = "placebo",
                sm = "OR", comb.fixed = FALSE)
#
summary(net1)
forest(net1, xlim = c(0.2, 50))

# Additive model for treatment components (with placebo as inactive
# treatment)
#
nc1 <- netcomb(net1, inactive = "placebo")
#
summary(nc1)
forest(nc1, xlim = c(0.2, 50))

## Not run:
# Specify, order of treatments
#
trts <- c("TCA", "SSRI", "SNRI", "NRI", "Low-dose SARI", "NaSSa",
          "rMAO-A", "Ind drug", "Hypericum", "Face-to-face CBT",
          "Face-to-face PST", "Face-to-face interpsy", "Face-to-face psychodyn",
          "Other face-to-face", "Remote CBT", "Self-help CBT", "No contact CBT",
          "Face-to-face CBT + SSRI", "Face-to-face interpsy + SSRI",
```

```
          "Face-to-face PST + SSRI", "UC", "Placebo")
#
# Note, three treatments are actually combinations of 'SSRI' with
# other components:
#
# "Face-to-face CBT + SSRI",
# "Face-to-face interpsy + SSRI",
# "Face-to-face PST + SSRI"

# Conduct random effects network meta-analysis
#
net1 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = Linde2016, ref = "placebo",
                seq = trts,
                sm = "OR", comb.fixed = FALSE)
#
summary(net1)
forest(net1, xlim = c(0.2, 50))

# Additive model for treatment components (with placebo as inactive
# treatment)
#
nc1 <- netcomb(net1, inactive = "placebo")
#
summary(nc1)
forest(nc1, xlim = c(0.2, 50))

## End(Not run)
```

`netconnection` *Get information on network connectivity (number of subnetworks, distance matrix)*

Description

To determine the network structure and to test whether a given network is fully connected. Network information is provided as a triple of vectors `treat1`, `treat2`, and `studlab` where each row corresponds to an existing pairwise treatment comparison (`treat1`, `treat2`) in a study (`studlab`). The function calculates the number of subnetworks (connectivity components; value of 1 corresponds to a fully connected network) and the distance matrix (in block-diagonal form in the case of subnetworks). If some treatments are combinations of other treatments or have common components, an analysis based on the additive network meta-analysis model might be possible, see discomb function.

Usage

```
netconnection(treat1, treat2, studlab, data = NULL, subset = NULL,
              nchar.trts = 666, title = "", warn = FALSE)
```

```
## S3 method for class 'netconnection'
print(x, digits = max(4, .Options$digits - 3),
      nchar.trts = x$nchar.trts, ...)
```

Arguments

treat1	Label/Number for first treatment.
treat2	Label/Number for second treatment.
studlab	An optional - but important! - vector with study labels (see Details).
data	An optional data frame containing the study information.
subset	An optional vector specifying a subset of studies to be used.
title	Title of meta-analysis / systematic review.
nchar.trts	A numeric defining the minimum number of characters used to create unique treatment names.
warn	A logical indicating whether warnings should be printed.
x	An object of class netconnection.
digits	Minimal number of significant digits, see print.default.
...	Additional arguments (ignored at the moment)

Value

An object of class netconnection with corresponding print function. The object is a list containing the following components:

treat1, treat2, studlab, title, warn, nchar.trts	As defined above.
k	Total number of studies.
m	Total number of pairwise comparisons.
n	Total number of treatments.
n.subnets	Number of subnetworks; equal to 1 for a fully connected network.
D.matrix	Distance matrix.
A.matrix	Adjacency matrix.
L.matrix	Laplace matrix.
call	Function call.
version	Version of R package netmeta used to create object.

Author(s)

Gerta Rücker <ruecker@imbi.uni-freiburg.de>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

netmeta, netdistance, discomb

Examples

```
data(Senn2013)

nc1 <- netconnection(treat1, treat2, studlab, data = Senn2013)
nc1

# Extract number of (sub)networks
nc1$n.subnets

# Extract distance matrix
nc1$D.matrix

# Conduct network meta-analysis (results not shown)
net1 <- netmeta(TE, seTE, treat1, treat2, studlab, data = Senn2013)


#
# Artificial example with two subnetworks
#
t1 <- c("G", "B", "B", "D", "A", "F")
t2 <- c("B", "C", "E", "E", "H", "A")
#
nc2 <- netconnection(t1, t2)
nc2

# Number of subnetworks
nc2$n.subnets

# Extract distance matrix
nc2$D.matrix

# Conduct network meta-analysis
# (results in an error message due to unconnected network)
try(net2 <- netmeta(1:6, 1:6, t1, t2, 1:6))

# Conduct network meta-analysis on first subnetwork
net2.1 <- netmeta(1:6, 1:6, t1, t2, 1:6,
                  subset = (t1 %in% c("A", "F", "H") & t2 %in% c("A", "F", "H")))

# Conduct network meta-analysis on first subnetwork
net2.2 <- netmeta(1:6, 1:6, t1, t2, 1:6,
                  subset = !(t1 %in% c("A", "F", "H") & t2 %in% c("A", "F", "H")))

summary(net2.1)
summary(net2.2)
```

`netdistance` *Calculate distance matrix for an adjacency matrix*

Description

Calculate distance matrix for an adjacency matrix based on distance algorithm by Müller et al. (1987).

Usage

```
netdistance(x)
```

Arguments

x	Either a netmeta object or an adjacency matrix.

Author(s)

Gerta Rücker <ruecker@imbi.uni-freiburg.de>

References

Müller WR, Szymanski K, Knop JV, and Trinajstic N (1987): An algorithm for construction of the molecular distance matrix. *Journal of Computational Chemistry*, **8**, 170–73

See Also

netmeta, netconnection

Examples

```
data(Senn2013)

net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data = Senn2013, sm = "MD")

netdistance(net1)
netdistance(net1$A.matrix)
```

---

netgraph *Network graph*

Description

This function generates a graph of the evidence network.

Usage

```
netgraph(x, seq = x$seq,
         labels = x$trts,
         cex = 1, adj = NULL,
         offset = if (!is.null(adj) && all(unique(adj) == 0.5)) 0 else 0.0175,
         scale = 1.10,
         col = "slateblue", plastic, thickness,
         lwd = 5, lwd.min = lwd/2.5, lwd.max = lwd*4,
         dim = "2d",
         highlight = NULL, col.highlight = "red2", lwd.highlight = lwd,
         multiarm = any(x$narms > 2), col.multiarm = NULL,
         alpha.transparency = 0.5,
         points = FALSE, col.points = "red", cex.points = 1, pch.points = 20,
         number.of.studies = FALSE,
         cex.number.of.studies = cex,
         col.number.of.studies = "white",
         bg.number.of.studies = "black",
         pos.number.of.studies = 0.5,
         start.layout = ifelse(dim  == "2d", "circle", "eigen"),
         eig1 = 2, eig2 = 3, eig3 = 4,
         iterate, tol = 0.0001, maxit = 500, allfigures = FALSE,
         A.matrix = x$A.matrix, N.matrix = sign(A.matrix),
         D.matrix = netdistance(N.matrix),
         xpos = NULL, ypos = NULL, zpos = NULL,
         figure = TRUE, ...)
```

Arguments

x	An object of class netmeta (mandatory).
seq	A character or numerical vector specifying the sequence of treatments arrangement (anticlockwise if start.layout = "circle").
labels	An optional vector with treatment labels.
cex	The magnification to be used for treatment labels.
col	A single color (or vector of colors) for lines connecting treatments (edges) if argument plastic = FALSE. Length of the vector must be equal to the number of edges.
adj	One, two, or three values in [0, 1] (or a vector / matrix with length / number of rows equal to the number of treatments) specifying the x (and optionally y and z) adjustment for treatment labels.
offset	Distance between edges (i.e. treatments) in graph and treatment labels for 2-D plots (value of 0.0175 corresponds to a difference of 1.75% of the range on x- and y-axis).
scale	Additional space added outside of edges (i.e. treatments). Increase this value for larger treatment labels (value of 1.10 corresponds to an additional space of 10% around the network graph).

`plastic` A logical indicating whether the appearance of the comparisons should be in '3D look' (not to be confused with argument `dim`).

`thickness` Either a character variable to determine the method to plot line widths (see Details) or a matrix of the same dimension and row and column names as argument `A.matrix` with information on line width.

`lwd` A numeric for scaling the line width of comparisons.

`lwd.max` Maximum line width in network graph. The connection with the largest value according to argument `thickness` will be set to this value.

`lwd.min` Minimum line width in network graph. All connections with line widths below this values will be set to `lwd.min`.

`dim` A character string indicating whether a 2- or 3-dimensional plot should be produced, either `"2d"` or `"3d"`.

`highlight` A character vector identifying comparisons that should be marked in the network graph, e.g. `highlight = "treat1:treat2"`.

`col.highlight` Color for highlighting the comparisons given by `highlight`.

`lwd.highlight` A numeric for the line width for highlighting the comparisons given by `highlight`.

`multiarm` A logical indicating whether multi-arm studies should be marked in plot.

`col.multiarm` Either a function from R library colorspace or grDevice to define colors for multi-arm studies or a character vector with colors to highlight multi-arm studies.

`alpha.transparency`
The alpha transparency of colors used to highlight multi-arm studies (0 means transparent and 1 means opaque).

`points` A logical indicating whether points should be printed at nodes (i.e. treatments) of the network graph.

`col.points, cex.points, pch.points`
Corresponding color, size, type for points. Can be a vector with length equal to the number of treatments.

`number.of.studies`
A logical indicating whether number of studies should be added to network graph.

`cex.number.of.studies`
The magnification to be used for number of studies.

`col.number.of.studies`
Color for number of studies.

`bg.number.of.studies`
Color for shadow around number of studies.

`pos.number.of.studies`
A single value (or vector of values) in [0, 1] specifying the position of the number of studies on the lines connecting treatments (edges). Length of the vector must be equal to the number of edges.

`start.layout` A character string indicating which starting layout is used if `iterate = TRUE`. If "circle" (default), the iteration starts with a circular ordering of the vertices; if "eigen", eigenvectors of the Laplacian matrix are used, calculated via generic

	function eigen (spectral decomposition); if "prcomp", eigenvectors of the Laplacian matrix are calculated via generic function prcomp (principal component analysis); if "random", a random layout is used, drawn from a bivariate normal.
eig1	A numeric indicating which eigenvector is used as x coordinate if start = "eigen" or "prcomp" and iterate = TRUE. Default is 2, the eigenvector to the second-smallest eigenvalue of the Laplacian matrix.
eig2	A numeric indicating which eigenvector is used as y-coordinate if start = "eigen" or "prcomp" and iterate = TRUE. Default is 3, the eigenvector to the third-smallest eigenvalue of the Laplacian matrix.
eig3	A numeric indicating which eigenvector is used as z-coordinate if start = "eigen" or "prcomp" and iterate = TRUE. Default is 4, the eigenvector to the fourth-smallest eigenvalue of the Laplacian matrix.
iterate	A logical indicating whether the stress majorization algorithm is carried out for optimization of the layout.
tol	A numeric for the tolerance for convergence if iterate = TRUE.
maxit	An integer defining the maximum number of iteration steps if iterate = TRUE.
allfigures	A logical indicating whether all iteration steps are shown if iterate = TRUE. May slow down calculations if set to TRUE (especially if plastic = TRUE).
A.matrix	Adjacency matrix (nxn) characterizing the structure of the network graph. Row and column names must be the same set of values as provided by argument seq.
N.matrix	Neighborhood matrix (nxn) replacing A.matrix if neighborhood is to be specified differently from node adjacency in the network graph, for example content-based. Row and column names must be the same set of values as provided by argument seq.
D.matrix	Distance matrix (nxn) replacing A.matrix and N.matrix if distances should be provided directly. Row and column names must be the same set of values as provided by argument seq.
xpos	Vector (n) of x coordinates.
ypos	Vector (n) of y coordinates.
zpos	Vector (n) of z coordinates.
figure	A logical indicating whether network graph should be shown.
...	Additional graphical arguments.

Details

The network is laid out in the plane, where the nodes in the graph layout correspond to the treatments and edges display the observed treatment comparisons. For the default setting, nodes are placed on a circle. Other starting layouts are "eigen", "prcomp", and "random" (Rücker & Schwarzer 2015). If iterate = TRUE, the layout is further optimized using the stress majorization algorithm. This algorithm specifies an 'ideal' distance (e.g., the graph distance) between two nodes in the plane. In the optimal layout, these distances are best approximated in the sense of least squares. Starting from an initial layout, the optimum is approximated in an iterative process called stress majorization (Kamada and Kawai 1989, Michailidis and de Leeuw 2001, Hu 2012). The starting layout can be chosen as a circle or coming from eigenvectors of the Laplacian matrix (corresponding to Hall's

algorithm, Hall 1970), calculated in different ways, or random. Moreover, it can be chosen whether the iteration steps are shown (argument `allfigures = TRUE`).

An optimized circular presentation which typically has a reduced (sometimes minimal) number of crossings can be achieved by using argument `seq = "optimal"` in combination with argument `start.layout`. Note, is is not possible of prespecify the best value for argument `start.layout` for any situation as the result depends on the network structure.

Argument `thickness` providing the line width of the nodes (comparisons) can be a matrix of the same dimension as argument `A.matrix` or any of the following character variables:

- Same line width (argument `lwd`) for all comparisons (`thickness = "equal"`)
- Proportional to number of studies comparing two treatments (`thickness = "number.of.studies"`)
- Proportional to inverse standard error of fixed effect model comparing two treatments (`thickness = "se.fixed"`)
- Proportional to inverse standard error of random effects model comparing two treatments (`thickness = "se.random"`)
- Weight from fixed effect model comparing two treatments (`thickness = "w.fixed"`)
- Weight from random effects model comparing two treatments (`thickness = "w.random"`)

Only evidence from direct treatment comparisons is considered to determine the line width if argument `thickness` is equal to any but the first method. By default, `thickness = "se.fixed"` is used if `start.layout = "circle"`, `iterate = FALSE`, and `plastic = TRUE`. Otherwise, the same line width is used.

Further, a couple of graphical parameters can be specified, such as color and appearance of the edges (treatments) and the nodes (comparisons), whether special comparisons should be highlighted and whether multi-arm studies should be indicated as colored polygons. By default, if R package colorspace is available the `sequential_hcl` function is used to highlight multi-arm studies; otherwise the `rainbow` is used.

In order to generate 3-D plots (argument `dim = "3d"`), R package **rgl** is necessary. Note, under macOS the X.Org X Window System must be available (see `https://www.xquartz.org`).

Value

A data frame containing the following columns

`labels`	Treatment labels.
`seq`	Sequence of treatment labels.
`xpos`	Position of treatment / edge on x-axis.
`ypos`	Position of treatment / edge on y-axis.
`zpos`	Position of treatment / edge on z-axis (for 3-D plots).
`xpos.labels`	Position of treatment labels on x-axis (for 2-D plots).
`ypos.labels`	Position of treatment labels on y-axis (for 2-D plots).
`adj.x`	Adjustment for treatment label on x-axis.
`adj.y`	Adjustment for treatment label on y-axis.
`adj.z`	Adjustment for treatment label on z-axis (for 3-D plots).

Author(s)

Gerta Rücker <ruecker@imbi.uni-freiburg.de>, Ulrike Krahn <ulrike.krahn@bayer.com>, Jochem König <koenigjo@uni-mainz.de>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Hall KM (1970): An r-dimensional quadratic placement algorithm. *Management Science*, **17**, 219–29

Hu Y (2012): *Combinatorial Scientific Computing*, Chapter Algorithms for Visualizing Large Networks, pages 525–49 Chapman and Hall/CRC Computational Science.

Kamada T, Kawai S (1989): An algorithm for drawing general undirected graphs. *Information Processing Letters*, **31**, 7–15

Krahn U, Binder H, König J (2013): A graphical tool for locating inconsistency in network meta-analyses. *BMC Medical Research Methodology*, **13**, 35

Michailidis G, de Leeuw J (2001): Data visualization through graph drawing. *Computational Statistics*, **16**, 435–50

Rücker G, Schwarzer G (2016): Automated drawing of network plots in network meta-analysis. *Research Synthesis Methods*, **7**, 94–107

See Also

netmeta

Examples

```
data(Senn2013)

# Generation of an object of class 'netmeta' with reference treatment 'plac'
#
net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
        data = Senn2013, sm = "MD", reference = "plac")

# Network graph with default settings
#
netgraph(net1)

# Network graph with specified order of the treatments and one
# highlighted comparison
#
trts <- c("plac", "benf", "migl", "acar", "sulf",
          "metf", "rosi", "piog", "sita", "vild")
netgraph(net1, highlight = "rosi:plac", seq = trts)

# Same network graph using argument 'seq' in netmeta function
#
net2 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data = Senn2013, sm = "MD", reference = "plac",
                seq = trts)
netgraph(net2, highlight = "rosi:plac")
```

```
# Network graph optimized, starting from a circle, with multi-arm
# study colored
#
netgraph(net1, start = "circle", iterate = TRUE, col.multiarm = "purple")

# Network graph optimized, starting from a circle, with multi-arm
# study colored and all intermediate iteration steps visible
#
## Not run: netgraph(net1, start = "circle", iterate = TRUE, col.multiarm = "purple",
         allfigures = TRUE)
## End(Not run)

# Network graph optimized, starting from Laplacian eigenvectors, with
# multi-arm study colored
#
netgraph(net1, start = "eigen", col.multiarm = "purple")

# Network graph optimized, starting from different Laplacian
# eigenvectors, with multi-arm study colored
#
netgraph(net1, start = "prcomp", col.multiarm = "purple")

# Network graph optimized, starting from random initial layout, with
# multi-arm study colored
#
netgraph(net1, start = "random", col.multiarm = "purple")

# Network graph without plastic look and one highlighted comparison
#
netgraph(net1, plastic = FALSE, highlight = "rosi:plac")

# Network graph without plastic look and comparisons with same
# thickness
netgraph(net1, plastic = FALSE, thickness = FALSE)

# Network graph with changed labels and specified order of the
# treatments
#
netgraph(net1, seq = c(1, 3, 5, 2, 9, 4, 7, 6, 8, 10),
         labels = LETTERS[1:10])

## Not run:
# Network graph in 3-D (opens a new device, where you may rotate and
# zoom the plot using the mouse / the mouse wheel).
# The rgl package must be installed for 3-D plots.
#
netgraph(net1, dim = "3d")

## End(Not run)
```

`netheat` *Net heat plot*

Description

This function creates a net heat plot, a graphical tool for locating inconsistency in network meta-analyses.

Usage

```
netheat(x, random=FALSE, tau.preset=NULL, showall=FALSE,
        nchar.trts = x$nchar.trts, ...)
```

Arguments

`x`	An object of class `netmeta`.
`random`	A logical indicating whether the net heat plot should be based on a random effects model.
`tau.preset`	An optional value for the square-root of the between-study variance tau^2 for a random effects model on which the net heat plot will be based.
`showall`	A logical indicating whether results should be shown for all designs or only a sensible subset, see Details.
`nchar.trts`	A numeric defining the minimum number of characters used to create unique treatment names.
`...`	Additional arguments.

Details

The net heat plot is a matrix visualization proposed by Krahn et al. (2013) that highlights hot spots of inconsistency between specific direct evidence in the whole network and renders transparent possible drivers.

In this plot, the area of a gray square displays the contribution of the direct estimate of one design in the column to a network estimate in a row. In combination, the colors show the detailed change in inconsistency when relaxing the assumption of consistency for the effects of single designs. The colors on the diagonal represent the inconsistency contribution of the corresponding design. The colors on the off-diagonal are associated with the change in inconsistency between direct and indirect evidence in a network estimate in the row after relaxing the consistency assumption for the effect of one design in the column. Cool colors indicate an increase and warm colors a decrease: the stronger the intensity of the color, the greater the difference between the inconsistency before and after the detachment. So, a blue colored element indicates that the evidence of the design in the column supports the evidence in the row. A clustering procedure is applied to the heat matrix in order to find warm colored hot spots of inconsistency. In the case that the colors of a column corresponding to design d are identical to the colors on the diagonal, the detaching of the effect of design d dissolves the total inconsistency in the network.

The pairwise contrasts corresponding to designs of three- or multi-arm studies are marked by '_' following the treatments of the design.

By default (`showall=FALSE`), designs where only one treatment is involved in other designs of the network or where the removal of corresponding studies would lead to a splitting of the network do not contribute to the inconsistency assessment and are not incorporated into the net heat plot.

In the case of `random=TRUE`, the net heat plot is based on a random effects model generalised for multivariate meta-analysis in which the between-study variance tau^2 is estimated by the method of moments (see Jackson et al., 2012) and embedded in a full design-by-treatment interaction model (see Higgins et al., 2012).

Author(s)

Ulrike Krahn <ulrike.krahn@bayer.com>

References

Krahn U, Binder H, König J (2013): A graphical tool for locating inconsistency in network meta-analyses. *BMC Medical Research Methodology*, **13**, 35

Jackson D, White IR and Riley RD (2012): Quantifying the impact of between-study heterogeneity in multivariate meta-analyses. *Statistics in Medicine*, **31**, 3805–20

Higgins JPT, Jackson D, Barrett JK, Lu G, Ades AE, White IR (2012): Consistency and inconsistency in network meta-analysis: concepts and models for multi-arm studies. *Research Synthesis Methods*, **3**, 98–110

See Also

netmeta

Examples

```
data(Senn2013)

#
# Generation of an object of class 'netmeta' with
# reference treatment 'plac', i.e. placebo
#
net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD", reference="plac")

#
# Generate a net heat plot based on a fixed effects model
#
netheat(net1)

#
# Generate a net heat plot based on a random effects model
#
netheat(net1, random=TRUE)
```

netleague *Create and print league table for network meta-analysis results*

Description

A league table is a square matrix showing all pairwise comparisons in a network meta-analysis. Typically, both treatment estimates and confidence intervals are shown.

Usage

```
netleague(x, y,
          comb.fixed = x$comb.fixed, comb.random = x$comb.random,
          seq = x$seq, ci = TRUE, backtransf = TRUE,
          direct = FALSE,
          digits = gs("digits"),
          bracket = gs("CIbracket"),
          separator = gs("CIseparator"),
          text.NA = ".",
          big.mark = gs("big.mark"))

## S3 method for class 'netleague'
print(x,
      comb.fixed=x$comb.fixed, comb.random=x$comb.random, ...)
```

Arguments

x	An object of class netmeta or netleague (mandatory).
y	An object of class netmeta (optional).
comb.fixed	A logical indicating whether a league table should be printed for the fixed effects (common effects) network meta-analysis.
comb.random	A logical indicating whether a league table should be printed for the random effects network meta-analysis.
seq	A character or numerical vector specifying the sequence of treatments in rows and columns of a league table.
ci	A logical indicating whether confidence intervals should be shown.
backtransf	A logical indicating whether printed results should be back transformed. If backtransf=TRUE, results for sm="OR" are printed as odds ratios rather than log odds ratios, for example.
direct	A logical indicating whether league table with network estimates (default) or estimates from direct comparisons should be generated if argument y is not missing.
digits	Minimal number of significant digits, see print.default.
bracket	A character with bracket symbol to print lower confidence interval: "[", "(", "{", "".

separator	A character string with information on separator between lower and upper confidence interval.
text.NA	A character string to label missing values.
big.mark	A character used as thousands separator.
...	Additional arguments (ignored at the moment).

Details

A league table is a square matrix showing all pairwise comparisons in a network meta-analysis. Typically, both treatment estimates and confidence intervals are shown.

If argument y is not provided, the league table contains the network estimates from network meta-analysis object x in the lower triangle and the direct treatment estimates from pairwise comparisons in the upper triangle.

If argument y is provided, the league table contains information on treatment comparisons from network meta-analysis object x in the lower triangle and from network meta-analysis object y in the upper triangle. This is, for example, useful to print information on efficacy and safety in the same league table.

This implementation reports pairwise comparisons of the treatment in the row versus the treatment in the column in the lower triangle and column versus row in the upper triangle. This is a common presentation for network meta-analyses which allows to easily compare direction and magnitude of treatment effects. For example, given treatments A, B, and C, the results reported in the first row and second column as well as second row and first column are from the pairwise comparison A versus B. Note, this presentation is different from the printout of a network meta-analysis object which reports opposite pairwise comparisons in the lower and upper triangle, e.g., A versus B in the first row and second column and B versus A in the second row and first column.

If the same network meta-analysis object is used for arguments x and y, reciprocal treatment estimates will be shown in the upper triangle (see examples), e.g., the comparison B versus A.

R function netrank can be used to change the order of rows and columns in the league table (see examples).

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>, Gerta Rücker <ruecker@imbi.uni-freiburg.de>

See Also

netmeta, netposet, netrank

Examples

```
# Network meta-analysis of count mortality statistics
#
data(Woods2010)

p0 <- pairwise(treatment, event = r, n = N,
               studlab = author, data = Woods2010, sm = "OR")
net0 <- netmeta(p0)
```

```
oldopts <- options(width = 100)

# League table for fixed and random effects model with
# - network estimates in lower triangle
# - direct estimates in upper triangle
#
netleague(net0, digits = 2, bracket = "(", separator = " - ")

# League table for fixed effects model
#
netleague(net0, comb.random = FALSE, digits = 2)

# Change order of treatments according to treatment ranking
# (random effects model)
#
netleague(net0, comb.fixed = FALSE, digits = 2,
          seq = netrank(net0))
#
print(netrank(net0), comb.fixed = FALSE)

## Not run:
# Create a CSV file with league table for random effects model
#
league0 <- netleague(net0, digits = 2, bracket = "(", separator = " to ")
#
write.table(league0$random, file = "league0-random.csv",
            row.names = FALSE, col.names = FALSE,
            sep = ",")
#
# Create Excel files with league tables (using R package WriteXLS)
#
library(WriteXLS)
#
# League table from random effects model
#
WriteXLS(league0$random, ExcelFileName = "league0-random.xls",
         SheetNames = "leaguetable (random)", col.names = FALSE)
#
# League tables from fixed and random effects models
#
WriteXLS(list(league0$fixed, league0$random),
         ExcelFileName = "league0-both.xls",
         SheetNames = c("leaguetable (fixed)", "leaguetable (random)"),
         col.names = FALSE)

## End(Not run)

# Use depression dataset
#
data(Linde2015)
#
# Define order of treatments
```

```
#
trts <- c("TCA", "SSRI", "SNRI", "NRI",
          "Low-dose SARI", "NaSSa", "rMAO-A", "Hypericum",
          "Placebo")
#
# Outcome labels
#
outcomes <- c("Early response", "Early remission")
#
# (1) Early response
#
p1 <- pairwise(treat = list(treatment1, treatment2, treatment3),
               event = list(resp1, resp2, resp3),
               n = list(n1, n2, n3),
               studlab = id, data = Linde2015, sm = "OR")
#
net1 <- netmeta(p1, comb.fixed = FALSE,
                seq = trts, ref = "Placebo")
#
# (2) Early remission
#
p2 <- pairwise(treat = list(treatment1, treatment2, treatment3),
               event = list(remi1, remi2, remi3),
               n = list(n1, n2, n3),
               studlab = id, data = Linde2015, sm = "OR")
#
net2 <- netmeta(p2, comb.fixed = FALSE,
                seq = trts, ref = "Placebo")

options(width = 200)
netleague(net1, digits = 2)

netleague(net1, digits = 2, ci = FALSE)
netleague(net2, digits = 2, ci = FALSE)

# League table for two outcomes with
# - network estimates of first outcome in lower triangle
# - network estimates of second outcome in upper triangle
#
netleague(net1, net2, digits = 2, ci = FALSE)

netleague(net1, net2, seq = netrank(net1, small = "bad"), ci = FALSE)
netleague(net1, net2, seq = netrank(net2, small = "bad"), ci = FALSE)

print(netrank(net1, small = "bad"))
print(netrank(net2, small = "bad"))


# Report results for network meta-analysis twice
#
netleague(net1, net1, seq = netrank(net1, small = "bad"), ci = FALSE,
          backtransf = FALSE)
netleague(net1, net1, seq = netrank(net1, small = "bad"), ci = FALSE,
```

```
          backtransf = FALSE, direct = TRUE)

options(oldopts)


## Not run:
# Generate a partial order of treatment rankings
#
np <- netposet(net1, net2, outcomes = outcomes, small.values = rep("bad",2))
hasse(np)
plot(np)

## End(Not run)
```

`netmatrix` *Create a matrix with additional information for pairwise comparisons*

Description

Auxiliary function to create a matrix with additional information for pairwise comparisons

Usage

```
netmatrix(x, var, levels, labels = levels,
          func = "mode", ties.method = "random")
```

Arguments

`x`	A `netmeta` object.
`var`	Variable with additional information.
`levels`	An optional vector of the values that `var` might have taken (see `factor`).
`labels`	An optional vector with labels for `var` (see `factor`).
`func`	A character string with the function name to summarize values within pairwise comparisons; see Details.
`ties.method`	A character string describing how ties are handled if `func="mode"`; see Details.

Details

For each pairwise comparison, unique values will be calculated for the variable `var` based on the argument `func`: "mode" (most common value), "min" (minimum value), "max", "mean", "median", and "sum". In order to determine the most common value, the argument `ties.method` can be used in the case of ties with "first" meaning that the first / smallest value will be selected; similar for "last" (last / largest value) and "random" (random selection).

Value

A matrix with the same row and column names as the adjacency matrix `x$A.matrix`.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

netmeta, netgraph

Examples

```
data(smokingcessation)

#
# Add variable with (fictious) risk of bias values
# with 1 = "low risk" and 2 = "high risk"
#
smokingcessation$rob <- rep(1:2, 12)

p1 <- pairwise(list(treat1, treat2, treat3),
               event=list(event1, event2, event3),
               n=list(n1, n2, n3),
               data=smokingcessation,
               sm="OR")

net1 <- netmeta(p1, comb.fixed = FALSE, ref = "A")

# Generate network graph with information on risk of bias
#
col.rob <- netmatrix(net1, rob, ties.method = "last",
                     levels = 1:2,
                     labels = c("green", "yellow"))
#
netgraph(net1, col = col.rob,
         plastic = FALSE, thickness = "number.of.studies", multi = FALSE)

n.trts <- net1$n.trts
labs <- paste(net1$trts, " (n=", n.trts, ")", sep = "")
#
netgraph(net1, col = col.rob,
         plastic = FALSE, thickness = "number.of.studies", multi = FALSE,
         points = TRUE, col.points = "blue",
         cex.points = 6 * sqrt(n.trts / max(n.trts)),
         labels = labs)
```

---

netmeasures *Measures for characterizing a network meta-analysis*

Description

This function provides measures for quantifying the direct evidence proportion, the mean path length and the minimal parallelism (the latter on aggregated and study level) of mixed treatment comparisons (network estimates) as well as the evidence flow per design, see König et al. (2013). These measures support the critical evaluation of the network meta-analysis results by rendering transparent the process of data pooling.

Usage

```
netmeasures(x,
            random = x$comb.random | !missing(tau.preset),
            tau.preset = x$tau.preset, warn = TRUE)
```

Arguments

x	An object of class netmeta.
random	A logical indicating whether random effects model should be used to calculate network measures.
tau.preset	An optional value for the square-root of the between-study variance τ^2.
warn	A logical indicating whether warnings should be printed.

Details

The direct evidence proportion gives the absolute contribution of direct effect estimates combined for two-arm and multi-arm studies to one network estimate.

Concerning indirectness, comparisons with a mean path length beyond two should be interpreted with particular caution, as more than two direct comparisons have to be combined serially on average.

Large indices of parallelism, either on study-level or on aggregated level, can be considered as supporting the validity of a network meta-analysis if there is only a small amount of heterogeneity.

The network estimates for two treatments are linear combinations of direct effect estimates comparing these or other treatments. The linear coefficients can be seen as the generalization of weights known from classical meta-analysis. These coefficients are given in the projection matrix H of the underlying model. For multi-arm studies, the coefficients depend on the choice of the study-specific baseline treatment, but the absolute flow of evidence can be made explicit for each design as shown in König et al. (2013) and is given in H.tilde.

All measures are calculated based on the fixed effects meta-analysis by default. In the case that in function netmeta the argument comb.random=TRUE, all measures are calculated for a random effects model. The value of the square-root of the between-study variance tau^2 can also be prespecified by argument tau.preset in function netmeta.

Value

A list containing the following components:

random, tau.preset
: As defined above.

proportion	A named vector of the direct evidence proportion of each network estimate.
meanpath	A named vector of the mean path length of each network estimate.
minpar	A named vector of the minimal parallelism on aggregated level of each network estimate.
minpar.study	A named vector of the minimal parallelism on study level of each network estimate.
H.tilde	Design-based hat matrix with information on absolute evidence flow per design. The number of rows is equal to the number of possible pairwise treatment comparisons and the number of columns is equal to the number of designs.

Author(s)

Ulrike Krahn <ulrike.krahn@bayer.com>, Jochem König <koenigjo@uni-mainz.de>

References

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

netmeta

Examples

```
data(Senn2013)

#
# Conduct fixed effects network meta-analysis with
# reference treatment 'plac', i.e. placebo
#
net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD", reference="plac",
                comb.random=FALSE)

#
# Calculate measures based on a fixed effects model
#
nm1 <- netmeasures(net1)

#
# Plot of minimal parallelism versus mean path length
#
plot(nm1$meanpath, nm1$minpar, pch="",
     xlab="Mean path length", ylab="Minimal parallelism")
text(nm1$meanpath, nm1$minpar, names(nm1$meanpath), cex=0.8)

# Conduct random effects network meta-analysis with
# reference treatment 'plac', i.e. placebo
#
```

```
net2 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD", reference="plac",
                comb.fixed=FALSE)

#
# Calculate measures based on a random effects model
#
nm2 <- netmeasures(net2)
```

`netmeta` *Network meta-analysis using graph-theoretical method*

Description

Network meta-analysis is a generalisation of pairwise meta-analysis that compares all pairs of treatments within a number of treatments for the same condition. The graph-theoretical method for analysis of network meta-analyses uses graph-theoretical methods that were originally developed in electrical network theory. It has been found to be equivalent to the frequentist approach to network meta-analysis (Rücker, 2012).

Usage

```
netmeta(TE, seTE, treat1, treat2, studlab, data=NULL, subset=NULL,
        sm, level=gs("level"), level.comb=gs("level.comb"),
        comb.fixed=gs("comb.fixed"),
        comb.random=gs("comb.random") | !is.null(tau.preset),
        prediction=FALSE, level.predict=gs("level.predict"),
        reference.group="", baseline.reference=TRUE,
        all.treatments=NULL, seq=NULL, tau.preset=NULL,
        tol.multiarm = 0.0005, details.chkmultiarm = FALSE,
        sep.trts=":", nchar.trts=666,
        n1=NULL, n2=NULL, event1=NULL, event2=NULL,
        backtransf=gs("backtransf"), title="",
        keepdata=gs("keepdata"), warn=TRUE)
```

Arguments

`TE`	Estimate of treatment effect, i.e. difference between first and second treatment (e.g. log odds ratio, mean difference, or log hazard ratio).
`seTE`	Standard error of treatment estimate.
`treat1`	Label/Number for first treatment.
`treat2`	Label/Number for second treatment.
`studlab`	An optional - but important! - vector with study labels (see Details).
`data`	An optional data frame containing the study information.
`subset`	An optional vector specifying a subset of studies to be used.

`sm` A character string indicating underlying summary measure, e.g., `"RD"`, `"RR"`, `"OR"`, `"ASD"`, `"HR"`, `"MD"`, `"SMD"`, or `"ROM"`.

`level` The level used to calculate confidence intervals for individual comparisons.

`level.comb` The level used to calculate confidence intervals for pooled estimates.

`comb.fixed` A logical indicating whether a fixed effects (common effects) network meta-analysis should be conducted.

`comb.random` A logical indicating whether a random effects network meta-analysis should be conducted.

`prediction` A logical indicating whether prediction intervals should be printed.

`level.predict` The level used to calculate prediction intervals for a new study.

`reference.group`
Reference treatment.

`baseline.reference`
A logical indicating whether results should be expressed as comparisons of other treatments versus the reference treatment (default) or vice versa. This argument is only considered if `reference.group` has been specified.

`all.treatments` A logical or `"NULL"`. If `TRUE`, matrices with all treatment effects, and confidence limits will be printed.

`seq` A character or numerical vector specifying the sequence of treatments in printouts.

`tau.preset` An optional value for manually setting the square-root of the between-study variance τ^2.

`tol.multiarm` A numeric for the tolerance for consistency of treatment estimates and corresponding variances in multi-arm studies which are consistent by design.

`details.chkmultiarm`
A logical indicating whether treatment estimates and / or variances of multi-arm studies with inconsistent results or negative multi-arm variances should be printed.

`sep.trts` A character used in comparison names as separator between treatment labels.

`backtransf` A logical indicating whether results should be back transformed in printouts and forest plots. If `backtransf=TRUE`, results for `sm="OR"` are presented as odds ratios rather than log odds ratios, for example.

`nchar.trts` A numeric defining the minimum number of characters used to create unique treatment names (see Details).

`n1` Number of observations in first treatment group.

`n2` Number of observations in second treatment group.

`event1` Number of events in first treatment group.

`event2` Number of events in second treatment group.

`title` Title of meta-analysis / systematic review.

`keepdata` A logical indicating whether original data (set) should be kept in netmeta object.

`warn` A logical indicating whether warnings should be printed (e.g., if studies are excluded from meta-analysis due to zero standard errors).

Details

Network meta-analysis using R package **netmeta** is described in detail in Schwarzer et al. (2015), Chapter 8.

Let *n* be the number of different treatments (nodes, vertices) in a network and let *m* be the number of existing comparisons (edges) between the treatments. If there are only two-arm studies, *m* is the number of studies. Let TE and seTE be the vectors of observed effects and their standard errors. Let W be the *m*x*m* diagonal matrix that contains the inverse variance 1/seTE^2.

The given comparisons define the network structure. Therefrom an *m*x*n* design matrix B (edge-vertex incidence matrix) is formed; for more precise information, see Rücker (2012). Moreover, the *n*x*n* Laplacian matrix L and its Moore-Penrose pseudoinverse L+ are calculated (both matrices play an important role in graph theory and electrical network theory). Using these matrices, the variances based on both direct and indirect comparisons can be estimated. Moreover, the hat matrix H can be estimated by **H = BL+B^tW = B(B^t W B)^+B^tW** and finally consistent treatment effects can be estimated by applying the hat matrix to the observed (potentially inconsistent) effects. H is a projection matrix which maps the observed effects onto the consistent (n-1)-dimensional subspace. This is the Aitken estimator (Senn et al., 2013). As in pairwise meta-analysis, the Q statistic measures the deviation from consistency. Q can be separated into parts for each pairwise meta-analysis and a part for remaining inconsistency between comparisons.

Often multi-arm studies are included in a network meta-analysis. In multi-arm studies, the treatment effects on different comparisons are not independent, but correlated. This is accounted for by reweighting all comparisons of each multi-arm study. The method is described in Rücker (2012) and Rücker and Schwarzer (2014).

Comparisons belonging to multi-arm studies are identified by identical study labels (argument `studlab`). It is therefore important to use identical study labels for all comparisons belonging to the same multi-arm study, e.g., study label "Willms1999" for the three-arm study in the data example (Senn et al., 2013). The function netmeta then automatically accounts for within-study correlation by reweighting all comparisons of each multi-arm study.

Data entry for this function is in *contrast-based* format, that is, data are given as contrasts (differences) between two treatments (argument `TE`) with standard error (argument `seTE`). In principle, meta-analysis functions from R package **meta**, e.g. `metabin` for binary outcomes or `metacont` for continuous outcomes, can be used to calculate treatment effects separately for each treatment comparison which is a rather tedious enterprise. If data are provided in *arm-based* format, that is, data are given for each treatment arm separately (e.g. number of events and participants for binary outcomes), a much more convenient way to transform data into contrast-based form is available. Function `pairwise` can automatically transform data with binary outcomes (using the `metabin` function from R package **meta**), continuous outcomes (`metacont` function), incidence rates (`metainc` function), and generic outcomes (`metagen` function). Additional arguments of these functions can be provided, e.g., to calculate Hedges' *g* or Cohen's *d* for continuous outcomes (see help page of function `pairwise`).

Note, all pairwise comparisons must be provided for a multi-arm study. Consider a multi-arm study of *p* treatments with known variances. For this study, treatment effects and standard errors must be provided for each of *p*(*p* - 1)/2 possible comparisons. For instance, a three-arm study contributes three pairwise comparisons, a four-arm study even six pairwise comparisons. Function `pairwise` automatically calculates all pairwise comparisons for multi-arm studies.

A simple random effects model assuming that a constant heterogeneity variance is added to each comparison of the network can be defined via a generalised methods of moments estimate of the

between-studies variance tau^2 (Jackson et al., 2012). This is added to the observed sampling variance seTE^2 of each comparison in the network (before appropriate adjustment for multi-arm studies). Then, as in standard pairwise meta-analysis, the procedure is repeated with the resulting enlarged standard errors.

Internally, both fixed effects and random effects models are calculated regardless of values choosen for arguments `comb.fixed` and `comb.random`. Accordingly, the network estimates for the random effects model can be extracted from component `TE.random` of an object of class `"netmeta"` even if argument `comb.random=FALSE`. However, all functions in R package **netmeta** will adequately consider the values for `comb.fixed` and `comb.random`. E.g. function `print.summary.netmeta` will not print results for the random effects model if `comb.random=FALSE`.

By default, treatment names are not abbreviated in printouts. However, in order to get more concise printouts, argument `nchar.trts` can be used to define the minimum number of characters for abbreviated treatment names (see `abbreviate`, argument `minlength`). R function `treats` is utilised internally to create abbreviated treatment names.

Names of treatment comparisons are created by concatenating treatment labels of pairwise comparisons using `sep.trts` as separator (see `paste`). These comparison names are used in the covariance matrices `Cov.fixed` and `Cov.random` and in some R functions, e.g, `decomp.design`. By default, a colon is used as the separator. If any treatment label contains a colon the following characters are used as separator (in consecutive order): `"-"`, `"_"`, `"/"`, `"+"`, `"."`, `"|"`, and `"*"`. If all of these characters are used in treatment labels, a corresponding error message is printed asking the user to specify a different separator.

Value

An object of class `netmeta` with corresponding `print`, `summary`, `forest`, and `netrank` functions. The object is a list containing the following components:

`studlab, treat1, treat2, TE, seTE`
: As defined above.

`seTE.adj`
: Standard error of treatment estimate, adjusted for multi-arm studies.

`n1, n2, event1, event2`
: As defined above.

`k`
: Total number of studies.

`m`
: Total number of pairwise comparisons.

`n`
: Total number of treatments.

`d`
: Total number of designs (corresponding to the unique set of treatments compared within studies).

`trts`
: Treatments included in network meta-analysis.

`k.trts`
: Number of studies evaluating a treatment.

`n.trts`
: Number of observations receiving a treatment (if arguments `n1` and `n2` are provided).

`events.trts`
: Number of events observed for a treatment (if arguments `event1` and `event2` are provided).

`studies`
: Study labels coerced into a factor with its levels sorted alphabetically.

`narms`
: Number of arms for each study.

`designs` Unique list of designs present in the network. A design corresponds to the set of treatments compared within a study.

`TE.nma.fixed, TE.nma.random`
A vector of length *m* of consistent treatment effects estimated by network meta-analysis (nma) (fixed effect / random effects model).

`seTE.nma.fixed, seTE.nma.random`
A vector of length *m* of effective standard errors estimated by network meta-analysis (fixed effect / random effects model).

`lower.nma.fixed, lower.nma.random`
A vector of length *m* of lower confidence interval limits for consistent treatment effects estimated by network meta-analysis (fixed effect / random effects model).

`upper.nma.fixed, upper.nma.random`
A vector of length *m* of upper confidence interval limits for the consistent treatment effects estimated by network meta-analysis (fixed effect /random effects model).

`leverage.fixed` A vector of length *m* of leverages, interpretable as factors by which variances are reduced using information from the whole network.

`w.fixed, w.random`
A vector of length *m* of weights of individual studies (fixed effect / random effects model).

`Q.fixed` A vector of length *m* of contributions to total heterogeneity / inconsistency statistic.

`TE.fixed, TE.random`
*n*x*n* matrix with estimated overall treatment effects (fixed effect / random effects model).

`seTE.fixed, seTE.random`
*n*x*n* matrix with standard errors (fixed effect / random effects model).

`lower.fixed, upper.fixed, lower.random, upper.random`
*n*x*n* matrices with lower and upper confidence interval limits (fixed effect / random effects model).

`zval.fixed, pval.fixed, zval.random, pval.random`
*n*x*n* matrices with z-value and p-value for test of overall treatment effect (fixed effect / random effects model).

`seTE.predict` *n*x*n* matrix with standard errors for prediction intervals.

`lower.predict, upper.predict`
*n*x*n* matrices with lower and upper prediction interval limits.

`prop.direct.fixed, prop.direct.random`
A named vector of the direct evidence proportion of each network estimate. (fixed effect / random effects model).

`TE.direct.fixed, TE.direct.random`
*n*x*n* matrix with estimated treatment effects from direct evidence (fixed effect / random effects model).

`seTE.direct.fixed, seTE.direct.random`
*n*x*n* matrix with estimated standard errors from direct evidence (fixed effect / random effects model).

`lower.direct.fixed, upper.direct.fixed, lower.direct.random, upper.direct.random`
: *n*x*n* matrices with lower and upper confidence interval limits from direct evidence (fixed effect / random effects model).

`zval.direct.fixed, pval.direct.fixed, zval.direct.random, pval.direct.random`
: *n*x*n* matrices with z-value and p-value for test of overall treatment effect from direct evidence (fixed effect / random effects model).

`TE.indirect.fixed, TE.indirect.random`
: *n*x*n* matrix with estimated treatment effects from indirect evidence (fixed effect / random effects model).

`seTE.indirect.fixed, seTE.indirect.random`
: *n*x*n* matrix with estimated standard errors from indirect evidence (fixed effect / random effects model).

`lower.indirect.fixed, upper.indirect.fixed, lower.indirect.random, upper.indirect.random`
: *n*x*n* matrices with lower and upper confidence interval limits from indirect evidence (fixed effect / random effects model).

`zval.indirect.fixed, pval.indirect.fixed, zval.indirect.random, pval.indirect.random`
: *n*x*n* matrices with z-value and p-value for test of overall treatment effect from indirect evidence (fixed effect / random effects model).

`Q`
: Overall heterogeneity / inconsistency statistic.

`df.Q`
: Degrees of freedom for test of heterogeneity / inconsistency.

`pval.Q`
: P-value for test of heterogeneity / inconsistency.

`I2`
: I-squared.

`tau`
: Square-root of between-study variance.

`Q.heterogeneity`
: Overall heterogeneity statistic.

`df.Q.heterogeneity`
: Degrees of freedom for test of overall heterogeneity.

`pval.Q.heterogeneity`
: P-value for test of overall heterogeneity.

`Q.inconsistency`
: Overall inconsistency statistic.

`df.Q.inconsistency`
: Degrees of freedom for test of overall inconsistency.

`pval.Q.inconsistency`
: P-value for test of overall inconsistency.

`Q.decomp`
: Data frame with columns 'treat1', 'treat2', 'Q', 'df' and 'pval.Q', providing heterogeneity statistics for each pairwise meta-analysis of direct comparisons.

`A.matrix`
: Adjacency matrix (*n*x*n*).

`B.matrix`
: Edge-vertex incidence matrix (*m*x*n*).

`L.matrix`
: Laplacian matrix (*n*x*n*).

`Lplus.matrix`
: Moore-Penrose pseudoinverse of the Laplacian matrix (*n*x*n*).

`Q.matrix`
: Matrix of heterogeneity statistics for pairwise meta-analyses, where direct comparisons exist (*n*x*n*).

`G.matrix` Matrix with variances and covariances of comparisons (*m*x*m*). G is defined as **BL+B^t**.

`H.matrix` Hat matrix (*m*x*m*), defined as **H=GW=BL+B^tW**.

`n.matrix` *n*x*n* matrix with number of observations in direct comparisons (if arguments `n1` and `n2` are provided).

`events.matrix` *n*x*n* matrix with number of events in direct comparisons (if arguments `event1` and `event2` are provided).

`P.fixed, P.random`
*n*x*n* matrix with direct evidence proportions (fixed effect / random effects model).

`Cov.fixed` Variance-covariance matrix (fixed effect model)

`Cov.random` Variance-covariance matrix (random effects model)

`sm, level, level.comb`
As defined above.

`comb.fixed, comb.random`
As defined above.

`prediction, level.predict`
As defined above.

`reference.group, baseline.reference, all.treatments`
As defined above.

`seq, tau.preset, tol.multiarm, details.chkmultiarm`
As defined above.

`sep.trts, nchar.trts`
As defined above.

`backtransf, title, warn`
As defined above.

`call` Function call.

`version` Version of R package netmeta used to create object.

Author(s)

Gerta Rücker <ruecker@imbi.uni-freiburg.de>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Jackson D, White IR, Riley RD (2012): Quantifying the impact of between-study heterogeneity in multivariate meta-analyses. *Statistics in Medicine*, **31**, 3805–20

Rücker G (2012): Network meta-analysis, electrical networks and graph theory. *Research Synthesis Methods*, **3**, 312–24

Rücker G, Schwarzer G (2014): Reduce dimension or reduce weights? Comparing two approaches to multi-arm studies in network meta-analysis. *Statistics in Medicine*, **33**, 4353–69

Schwarzer G, Carpenter JR, Rücker G (2015): *Meta-Analysis with R (Use-R!)*. Springer International Publishing, Switzerland

Senn S, Gavini F, Magrez D, Scheen A (2013): Issues in performing a network meta-analysis. *Statistical Methods in Medical Research*, **22**, 169–89

See Also

pairwise, forest.netmeta, netrank, metagen

Examples

```
data(Senn2013)

#
# Conduct fixed effects network meta-analysis
#
net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD",
                comb.random=FALSE)
net1
net1$Q.decomp

#
# Comparison with reference group
#
print(net1, reference="plac")

#
# Conduct random effects network meta-analysis
#
net2 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD",
                comb.fixed=FALSE)
net2

#
# Change printing order of treatments with placebo last and
# use long treatment names
#
trts <- c("acar", "benf", "metf", "migl", "piog",
          "rosi", "sita", "sulf", "vild", "plac")
net3 <- netmeta(TE, seTE, treat1.long, treat2.long, studlab,
                data=Senn2013, sm="MD", comb.fixed=FALSE,
                seq=trts, reference="Placebo")
print(summary(net3), digits=2)
```

---

netmetabin *Network meta-analysis of binary outcome data*

Description

Provides three models for the network meta-analysis of binary data (Mantel-Haenszel method, based on the non-central hypergeometric distribution, and the inverse variance method).

Usage

```
netmetabin(event1, n1, event2, n2,
           treat1, treat2, studlab,
           data = NULL, subset = NULL,
           sm,
           method = "MH",
           cc.pooled = FALSE,
           incr = gs("incr"),
           allincr = gs("allincr"), addincr = gs("addincr"),
           allstudies = gs("allstudies"),
           level=gs("level"), level.comb=gs("level.comb"),
           comb.fixed = gs("comb.fixed"),
           comb.random = method == "Inverse" &
             (gs("comb.random") | !is.null(tau.preset)),
           ##
           prediction=FALSE, level.predict=gs("level.predict"),
           ##
           reference.group = "",
           baseline.reference = TRUE,
           all.treatments = NULL,
           seq = NULL,
           ##
           tau.preset = NULL,
           ##
           tol.multiarm = 0.0005,
           details.chkmultiarm = FALSE,
           ##
           sep.trts = ":",
           nchar.trts = 666,
           ##
           backtransf = gs("backtransf"),
           ##
           title = "",
           keepdata = gs("keepdata"),
           warn = TRUE)
```

Arguments

event1	Number of events (first treatment).
n1	Number of observations (first treatment).
event2	Number of events (second treatment).
n2	Number of observations (second treatment)
treat1	Label/Number for first treatment.
treat2	Label/Number for second treatment.
studlab	An optional - but important! - vector with study labels (see Details).
data	An optional data frame containing the study information.

`subset` An optional vector specifying a subset of studies to be used.

`sm` A character string indicating underlying summary measure, i.e., `"RD"`, `"RR"`, `"OR"`, `"ASD"`.

`method` A character string indicating which method is to be used for pooling of studies. One of `"Inverse"`, `"MH"`, or `"NCH"`, can be abbreviated.

`cc.pooled` A logical indicating whether `incr` should be used as a continuity correction, when calculating the network meta-analysis estimates.

`incr` A numerical value which is added to each cell count, i.e., to the numbers of events and non-events, of all treatment arms in studies with zero events or non-events in any of the treatment arms ("continuity correction").

`allincr` A logical indicating whether `incr` should be added to each cell count of all studies if a continuity correction was used for at least one study (only considered if `method="Inverse"`). If FALSE (default), `incr` is used as continuity correction only for studies with zero events or zero non-events in any of the treatment arms.

`addincr` A logical indicating whether `incr` should be added to each cell count of all studies, irrespective of zero cell counts (only considered if `method="Inverse"`).

`allstudies` A logical indicating whether studies with zero events or non-events in all treatment arms should be included in an inverse variance meta-analysis (applies only if `method="Inverse"` and `sm` is equal to either `"RR"` or `"OR"`).

`level` The level used to calculate confidence intervals for individual studies.

`level.comb` The level used to calculate confidence intervals for pooled estimates.

`comb.fixed` A logical indicating whether a fixed effects (common effects) network meta-analysis should be conducted.

`comb.random` A logical indicating whether a random effects network meta-analysis should be conducted.

`prediction` A logical indicating whether a prediction interval should be printed (only considered if `method="Inverse"`).

`level.predict` The level used to calculate prediction interval for a new study (only considered if `method="Inverse"`).

`reference.group`
Reference treatment.

`baseline.reference`
A logical indicating whether results should be expressed as comparisons of other treatments versus the reference treatment (default) or vice versa. This argument is only considered if `reference.group` has been specified.

`all.treatments` A logical or `"NULL"`. If `TRUE`, matrices with all treatment effects, and confidence limits will be printed.

`seq` A character or numerical vector specifying the sequence of treatments in printouts.

`tau.preset` An optional value for manually setting the square-root of the between-study variance τ^2 (only considered if `method="Inverse"`).

`tol.multiarm` A numeric for the tolerance for consistency of treatment estimates and corresponding variances in multi-arm studies which are consistent by design (only considered if `method="Inverse"`).

`details.chkmultiarm`
: A logical indicating whether treatment estimates and / or variances of multi-arm studies with inconsistent results or negative multi-arm variances should be printed (only considered if `method="Inverse"`).

`sep.trts`
: A character used in comparison names as separator between treatment labels.

`nchar.trts`
: A numeric defining the minimum number of characters used to create unique treatment names (see Details).

`backtransf`
: A logical indicating whether results should be back transformed in printouts and forest plots. If `backtransf=TRUE`, results for `sm="OR"` are presented as odds ratios rather than log odds ratios, for example.

`title`
: Title of meta-analysis / systematic review.

`keepdata`
: A logical indicating whether original data (set) should be kept in netmeta object.

`warn`
: A logical indicating whether warnings should be printed (e.g., if studies are excluded from meta-analysis due to zero standard errors).

Details

This function implements three models for the network meta-analysis of binary data:

- The Mantel-Haenszel network meta-analysis model, as described in Efthimiou et al. (2018) (`method="MH"`);
- a network meta-analysis model using the non-central hypergeometric distribution with the Breslow approximation, as described in Stijnen et al. (2010) (`method="NCH"`),
- the inverse variance method for network meta-analysis (`method="Inverse"`), also provided by `netmeta`.

Comparisons belonging to multi-arm studies are identified by identical study labels (argument `studlab`). It is therefore important to use identical study labels for all comparisons belonging to the same multi-arm study.

Data entry for this function is in *contrast-based* format, that is, each line of the data corresponds to a single pairwise comparison between two treatments (arguments `treat1`, `treat2`, `event1`, `n1`, `event2`, and `n2`). If data are provided in *arm-based* format, that is, number of events and participants are given for each treatment arm separately, function `pairwise` can be used to transform the data to *contrast-based* format (see help page of function `pairwise`).

Note, all pairwise comparisons must be provided for a multi-arm study. Consider a multi-arm study of p treatments with known variances. For this study, the number of events and observations must be provided for each treatment, for each of $p(p - 1)/2$ possible comparisons in separate lines in the data. For instance, a three-arm study contributes three pairwise comparisons, a four-arm study even six pairwise comparisons. Function `pairwise` automatically calculates all pairwise comparisons for multi-arm studies.

For `method = "Inverse"`, both fixed effects and random effects models are calculated regardless of values choosen for arguments `comb.fixed` and `comb.random`. Accordingly, the network estimates for the random effects model can be extracted from component `TE.random` of an object of class `"netmeta"` even if argument `comb.random=FALSE`. However, all functions in R package **netmeta** will adequately consider the values for `comb.fixed` and `comb.random`. E.g. function `print.summary.netmeta` will not print results for the random effects model if `comb.random=FALSE`.

For method = "MH" and method = "NCH", only a fixed effects model is available.

By default, treatment names are not abbreviated in printouts. However, in order to get more concise printouts, argument nchar.trts can be used to define the minimum number of characters for abbreviated treatment names (see abbreviate, argument minlength). R function treats is utilised internally to create abbreviated treatment names.

Names of treatment comparisons are created by concatenating treatment labels of pairwise comparisons using sep.trts as separator (see paste). These comparison names are used in the covariance matrices Cov.fixed and Cov.random and in some R functions, e.g, decomp.design. By default, a colon is used as the separator. If any treatment label contains a colon the following characters are used as separator (in consecutive order): "-", "_", "/", "+", ".", "|", and "*". If all of these characters are used in treatment labels, a corresponding error message is printed asking the user to specify a different separator.

Value

An object of class netmetabin and netmeta with corresponding print, summary, forest, and netrank functions. The object is a list containing the following components:

studlab, treat1, treat2
: As defined above.

n1, n2, event1, event2
: As defined above.

TE
: Estimate of treatment effect, i.e. difference between first and second treatment (e.g. log odds ratio).

seTE
: Standard error of treatment estimate.

k
: Total number of studies.

m
: Total number of pairwise comparisons.

n
: Total number of treatments.

d
: Total number of designs (corresponding to the unique set of treatments compared within studies).

trts
: Treatments included in network meta-analysis.

k.trts
: Number of studies evaluating a treatment.

n.trts
: Number of observations receiving a treatment.

events.trts
: Number of events observed for a treatment.

studies
: Study labels coerced into a factor with its levels sorted alphabetically.

narms
: Number of arms for each study.

designs
: Unique list of designs present in the network. A design corresponds to the set of treatments compared within a study.

TE.fixed, seTE.fixed
: *n*x*n* matrix with estimated overall treatment effects and standard errors for fixed effects model.

lower.fixed, upper.fixed
: *n*x*n* matrices with lower and upper confidence interval limits for fixed effects model.

`zval.fixed, pval.fixed`
: *n*x*n* matrices with z-value and p-value for test of overall treatment effect under fixed effects model.

`TE.random, seTE.random`
: *n*x*n* matrix with estimated overall treatment effects and standard errors for random effects model (only available if `method="Inverse"`).

`lower.random, upper.random`
: *n*x*n* matrices with lower and upper confidence interval limits for random effects model (only available if `method="Inverse"`).

`zval.random, pval.random`
: *n*x*n* matrices with z-value and p-value for test of overall treatment effect under random effects model (only available if `method="Inverse"`).

`TE.direct.fixed, seTE.direct.fixed`
: *n*x*n* matrix with estimated treatment effects and standard errors from direct evidence under fixed effects model.

`lower.direct.fixed, upper.direct.fixed`
: *n*x*n* matrices with lower and upper confidence interval limits from direct evidence under fixed effects model.

`zval.direct.fixed, pval.direct.fixed`
: *n*x*n* matrices with z-value and p-value for test of overall treatment effect from direct evidence under fixed effects model.

`TE.direct.random, seTE.direct.random`
: *n*x*n* matrix with estimated treatment effects and standard errors from direct evidence under random effects model (only available if `method="Inverse"`).

`lower.direct.random, upper.direct.random`
: *n*x*n* matrices with lower and upper confidence interval limits from direct evidence under random effects model (only available if `method="Inverse"`).

`zval.direct.random, pval.direct.random`
: *n*x*n* matrices with z-value and p-value for test of overall treatment effect from direct evidence under random effects model (only available if `method="Inverse"`).

`Q`
: Overall heterogeneity / inconsistency statistic.

`df.Q`
: Degrees of freedom for test of heterogeneity / inconsistency.

`pval.Q`
: P-value for test of heterogeneity / inconsistency.

`I2`
: I-squared (only available if `method="Inverse"`).

`tau`
: Square-root of between-study variance (only available if `method="Inverse"`).

`A.matrix`
: Adjacency matrix (*n*x*n*).

`H.matrix`
: Hat matrix (*m*x*m*)

`n.matrix`
: *n*x*n* matrix with number of observations in direct comparisons.

`events.matrix`
: *n*x*n* matrix with number of events in direct comparisons.

`sm, method, level, level.comb`
: As defined above.

`incr, allincr, addincr, allstudies, cc.pooled`
: As defined above.

`comb.fixed, comb.random`
: As defined above.

`prediction, level.predict`
: As defined above.

`reference.group, baseline.reference, all.treatments`
: As defined above.

`seq, tau.preset, tol.multiarm, details.chkmultiarm`
: As defined above.

`sep.trts, nchar.trts`
: As defined above.

`backtransf, title, warn`
: As defined above.

`data`
: Data set (in contrast-based format).

`data.design`
: List with data in arm-based format (each list element corresponds to a single design).

`call`
: Function call.

`version`
: Version of R package netmeta used to create object.

Author(s)

Orestis Efthimiou <oremiou@gmail.com>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Efthimiou O, Rücker G, Schwarzer G, Higgins J, Egger M, Salanti G (2018): A Mantel-Haenszel model for network meta-analysis of rare events. *Manuscript submitted for publication*.

Senn S, Gavini F, Magrez D, Scheen A (2013): Issues in performing a network meta-analysis. *Statistical Methods in Medical Research*, **22**, 169–89

Stijnen T, Hamza TH, Ozdemir P (2010): Random effects meta-analysis of event outcome in the framework of the generalized linear mixed model with applications in sparse data. *Statistics in Medicine*, **29**, 3046–67

See Also

pairwise, netmeta

Examples

```
data(Dong2013)

# Only consider first ten studies (to reduce runtime of example)
#
first10 <- subset(Dong2013, id <= 10)

# Transform data from long arm-based format to contrast-based format.
# Argument 'sm' has to be used for odds ratio as summary measure; by
# default the risk ratio is used in the metabin function called
# internally.
p1 <- pairwise(treatment, death, randomized, studlab = id,
               data = first10, sm = "OR")
```

```
# Conduct Mantel-Haenszel network meta-analysis
# (without continuity correction)
nb1 <- netmetabin(p1, ref = "plac")
nb1

# Obtain the league table
netleague(nb1)

## Not run:
# Conduct Mantel-Haenszel network meta-analysis for the whole dataset
#
p2 <- pairwise(treatment, death, randomized, studlab = id,
               data = Dong2013, sm = "OR")
netmetabin(p2, ref = "plac")

# Conduct network meta-analysis using the non-central hypergeometric
# model (without continuity correction)
netmetabin(p2, ref = "plac", method = "NCH")

# Conduct Mantel-Haenszel network meta-analysis
# (with continuity correction of 0.5; include all studies)
netmetabin(p2, ref = "plac", cc.pooled = TRUE)


data(Gurusamy2011)

p3 <- pairwise(treatment, death, n, studlab = study,
               data = Gurusamy2011, sm = "OR")

# Conduct Mantel-Haenszel network meta-analysis
# (without continuity correction)
netmetabin(p3, ref = "cont")

## End(Not run)
```

`netposet` *Partial order of treatments in network meta-analysis*

Description

Partial order of treatments in network meta-analysis. The set of treatments in a network is called a partially ordered set (in short, a *poset*), if different outcomes provide different treatment ranking lists.

Usage

```
netposet(..., outcomes, treatments, small.values,
         comb.fixed, comb.random)

## S3 method for class 'netposet'
```

```
print(x,
      pooled=ifelse(x$comb.random, "random", "fixed"),
      ...)
```

Arguments

...	See details.
outcomes	A character vector with outcome names.
treatments	A character vector with treatment names.
small.values	See details.
comb.fixed	A logical indicating whether to show results for the fixed effects (common effects) model.
comb.random	A logical indicating whether to show results for the random effects model.
x	An object of class netposet.
pooled	A character string indicating whether Hasse diagram should be drawn for fixed effect ("fixed") or random effects model ("random"). Can be abbreviated.

Details

In network meta-analysis, frequently different outcomes are considered which may each provide a different ordering of treatments. The concept of a partially ordered set (in short, a *poset*, Carlsen & Bruggemann, 2014) of treatments can be used to gain further insights in situations with apparently conflicting orderings. This implementation for rankings in network meta-analyis is described in Rücker & Schwarzer (2017).

In function netposet, argument ... can be any of the following:

- arbitrary number of netrank objects providing P-scores;
- arbitrary number of netmeta objects;
- single ranking matrix with each column providing P-scores (Rücker & Schwarzer 2015) or SUCRA values (Salanti et al. 2011) for an outcome and rows corresponding to treatments.

Note, albeit in general a ranking matrix is not constrained to have values between 0 and 1, netposet stops with an error in this case as this function expects a matrix with P-scores or SUCRA values.

Argument outcomes can be used to label outcomes. If argument outcomes is missing,

- column names of the ranking matrix are used as outcome labels (if first argument is a ranking matrix and column names are available);
- capital letters 'A', 'B', ... are used as outcome labels and a corresponding warning is printed.

Argument treatments can be used to provide treatment labels if the first argument is a ranking matrix. If argument treatment is missing,

- row names of the ranking matrix are used as treatment labels (if available);
- letters 'a', 'b', ... are used as treatment labels and a corresponding warning is printed.

If argument ... consists of netmeta objects, netrank is called internally to calculate P-scores. In this case, argument small.values can be used to specify for each outcome whether small values are good or bad; see netrank. This argument is ignored for a ranking matrix and netrank objects.

Arguments comb.fixed and comb.random can be used to define whether results should be printed and plotted for fixed effect and / or random effects model. If netmeta and netrank objects are provided in argument ..., values for comb.fixed and comb.random within these objects are considered; if these values are not unique, argument comb.fixed and / or comb.random are set to TRUE.

In function print.netposet, argument ... is passed on to the printing function.

Value

An object of class netposet with corresponding print, plot, and hasse functions. The object is a list containing the following components:

P.fixed	Ranking matrix with rows corresponding to treatments and columns corresponding to outcomes (fixed effect model).
M0.fixed	Hasse matrix skipping unnecessary paths (fixed effect model).
M.fixed	"Full" Hasse matrix (fixed effect model).
O.fixed	Matrix with information about partial ordering (fixed effect model).
P.random	Ranking matrix with rows corresponding to treatments and columns corresponding to outcomes (random effects model).
M0.random	Hasse matrix skipping unnecessary paths (random effects model).
M.random	"Full" Hasse matrix (random effects model).
O.random	Matrix with information about partial ordering (random effects model).
small.values, comb.fixed, comb.random	As.defined above.
call	Function call.
version	Version of R package netmeta used to create object.

Author(s)

Gerta Rücker <ruecker@imbi.uni-freiburg.de>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

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

Rücker G, Schwarzer G (2015): Ranking treatments in frequentist network meta-analysis works without resampling methods. *BMC Medical Research Methodology*, **15**, 58

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

netmeta, netrank, hasse, plot.netposet

Examples

```
# Use depression dataset
#
data(Linde2015)
#
# Define order of treatments
#
trts <- c("TCA", "SSRI", "SNRI", "NRI",
          "Low-dose SARI", "NaSSa", "rMAO-A", "Hypericum",
          "Placebo")
#
# Outcome labels
#
outcomes <- c("Early response", "Early remission")
#
# (1) Early response
#
p1 <- pairwise(treat = list(treatment1, treatment2, treatment3),
               event = list(resp1, resp2, resp3),
               n = list(n1, n2, n3),
               studlab = id, data = Linde2015, sm = "OR")
#
net1 <- netmeta(p1, comb.fixed = FALSE,
                seq = trts, ref = "Placebo")
#
# (2) Early remission
#
p2 <- pairwise(treat = list(treatment1, treatment2, treatment3),
               event = list(remi1, remi2, remi3),
               n = list(n1, n2, n3),
               studlab = id, data = Linde2015, sm = "OR")
#
net2 <- netmeta(p2, comb.fixed = FALSE,
                seq = trts, ref = "Placebo")
#
# Partial order of treatment rankings (two outcomes)
#
po <- netposet(netrank(net1, small.values = "bad"),
               netrank(net2, small.values = "bad"),
               outcomes = outcomes)
#
# Hasse diagram
#
hasse(po)


## Not run:
#
```

```
# Outcome labels
#
outcomes <- c("Early response", "Early remission",
              "Lost to follow-up", "Lost to follow-up due to AEs",
              "Adverse events (AEs)")
#
# (3) Loss to follow-up
#
p3 <- pairwise(treat = list(treatment1, treatment2, treatment3),
               event = list(loss1, loss2, loss3),
               n = list(n1, n2, n3),
               studlab = id, data = Linde2015, sm = "OR")
#
net3 <- netmeta(p3, comb.fixed = FALSE,
                seq = trts, ref = "Placebo")
#
# (4) Loss to follow-up due to adverse events
#
p4 <- pairwise(treat = list(treatment1, treatment2, treatment3),
               event = list(loss.ae1, loss.ae2, loss.ae3),
               n = list(n1, n2, n3),
               studlab = id, data = subset(Linde2015, id != 55),
               sm = "OR")
#
net4 <- netmeta(p4, comb.fixed = FALSE,
                seq = trts, ref = "Placebo")
#
# (5) Adverse events
#
p5 <- pairwise(treat = list(treatment1, treatment2, treatment3),
               event = list(ae1, ae2, ae3),
               n = list(n1, n2, n3),
               studlab = id, data = Linde2015, sm = "OR")
#
net5 <- netmeta(p5, comb.fixed = FALSE,
                seq = trts, ref = "Placebo")
#
# Partial order of treatment rankings (all five outcomes)
#
po.ranks <- netposet(netrank(net1, small.values = "bad"),
                     netrank(net2, small.values = "bad"),
                     netrank(net3, small.values = "good"),
                     netrank(net4, small.values = "good"),
                     netrank(net5, small.values = "good"),
                     outcomes = outcomes)
#
# Same result
#
po.nets <- netposet(net1, net2, net3, net4, net5,
                    small.values = c("bad", "bad", "good", "good", "good"),
                    outcomes = outcomes)
#
all.equal(po.ranks, po.nets)
```

```
#
# Print matrix with P-scores (random effects model)
#
po.nets$P.random
#
# Hasse diagram for all outcomes (random effects model)
#
hasse(po.ranks)
#
# Hasse diagram for outcomes early response and early remission
#
po12 <- netposet(netrank(net1, small.values = "bad"),
                 netrank(net2, small.values = "bad"),
                 outcomes = outcomes[1:2])
hasse(po12)
#
# Scatter plot
#
oldpar <- par(pty = "s")
plot(po12)
par(oldpar)

## End(Not run)


# Example using ranking matrix with P-scores
#
# Ribassin-Majed L, Marguet S, Lee A.W., et al. (2017):
# What is the best treatment of locally advanced nasopharyngeal
# carcinoma? An individual patient data network meta-analysis.
# Journal of Clinical Oncology, 35, 498-505
#
outcomes <- c("OS", "PFS", "LC", "DC")
treatments <- c("RT", "IC-RT", "IC-CRT", "CRT",
                "CRT-AC", "RT-AC", "IC-RT-AC")
#
# P-scores (from Table 1)
#
pscore.os  <- c(15, 33, 63, 70, 96, 28, 45) / 100
pscore.pfs <- c( 4, 46, 79, 52, 94, 36, 39) / 100
pscore.lc  <- c( 9, 27, 47, 37, 82, 58, 90) / 100
pscore.dc  <- c(16, 76, 95, 48, 72, 32, 10) / 100
#
pscore.matrix <- data.frame(pscore.os, pscore.pfs, pscore.lc, pscore.dc)
rownames(pscore.matrix) <- treatments
colnames(pscore.matrix) <- outcomes
pscore.matrix
#
po <- netposet(pscore.matrix)
po12 <- netposet(pscore.matrix[, 1:2])
po
po12
#
```

```
hasse(po)
hasse(po12)
#
oldpar <- par(pty = "s")
plot(po12)
par(oldpar)
```

`netrank` *Frequentist method to rank treatments in network*

Description

Ranking treatments in frequentist network meta-analysis without resampling methods.

Usage

```
netrank(x, small.values="good")

## S3 method for class 'netrank'
print(x,
      comb.fixed = x$x$comb.fixed, comb.random = x$x$comb.random,
      sort=TRUE, digits=max(4, .Options$digits - 3), ...)
```

Arguments

`x`	An object of class `netmeta` (netrank function) or `netrank` (print function).
`comb.fixed`	A logical indicating whether to print P-scores for the fixed effects (common effects) model.
`comb.random`	A logical indicating whether to print P-scores for the random effects model.
`small.values`	A character string specifying whether small treatment effects indicate a beneficial (`"good"`) or harmful (`"bad"`) effect, can be abbreviated.
`sort`	A logical indicating whether printout should be sorted by decreasing P-score.
`digits`	Minimal number of significant digits, see `print.default`.
`...`	Additional arguments passed on to `print.data.frame` function (used internally).

Details

Treatments are ranked based on a network meta-analysis. Ranking is performed by P-scores. P-scores are based solely on the point estimates and standard errors of the network estimates. They measure the extent of certainty that a treatment is better than another treatment, averaged over all competing treatments (Rücker and Schwarzer 2015).

The P-score of treatment *i* is defined as the mean of all 1 - P[*j*] where P[*j*] denotes the one-sided P-value of accepting the alternative hypothesis that treatment *i* is better than one of the competing treatments *j*. Thus, if treatment *i* is better than many other treatments, many of these P-values will

be small and the P-score will be large. Vice versa, if treatment *i* is worse than most other treatments, the P-score is small.

The P-score of treatment *i* can be interpreted as the mean extent of certainty that treatment *i* is better than another treatment. This interpretation is comparable to that of the Surface Under the Cumulative RAnking curve (SUCRA) which is the rank of treatment *i* within the range of treatments, measured on a scale from 0 (worst) to 1 (best) (Salanti et al. 2011).

Value

An object of class `netrank` with corresponding `print` function. The object is a list containing the following components:

`Pscore.fixed`	A named numeric vector with P-scores for fixed effect model.
`Pmatrix.fixed`	Numeric matrix based on pairwise one-sided p-values for fixed effect model.
`Pscore.random`	A named numeric vector with P-scores for random effects model.
`Pmatrix.random`	Numeric matrix based on pairwise one-sided p-values of random effects model.
`small.values, x`	As defined above.
`version`	Version of R package netmeta used to create object.

Author(s)

Gerta Rücker <ruecker@imbi.uni-freiburg.de>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

Rücker G, Schwarzer G (2015): Ranking treatments in frequentist network meta-analysis works without resampling methods. *BMC Medical Research Methodology*, **15**, 58

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

`netmeta`

Examples

```
data(Senn2013)

net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD",
                comb.random=FALSE)
net2 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD",
comb.fixed=FALSE)
net3 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD")
```

```
nr1 <- netrank(net1)
nr1
print(nr1, sort=FALSE)

nr2 <- netrank(net2)
nr2
print(nr2, sort=FALSE)

nr3 <- netrank(net3)
nr3
print(nr3, sort="fixed")
print(nr3, sort=FALSE)
```

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

Description

Methods to split network estimates into the contribution of direct and indirect evidence and to test for local inconsistency in network meta-analysis.

Usage

```
netsplit(x, method, upper = TRUE,
         reference.group = x$reference.group,
         baseline.reference = x$baseline.reference,
         sep.trts = x$sep.trts, quote.trts = "",
         tol.direct = 0.0005, warn = FALSE)

## S3 method for class 'netsplit'
print(x,
      comb.fixed = x$comb.fixed,
      comb.random = x$comb.random,
      show = "all",
      overall = TRUE,
      ci = FALSE,
      test = show %in% c("all", "with.direct", "both"),
      digits = gs("digits"),
      digits.zval = gs("digits.zval"),
      digits.pval = gs("digits.pval"),
      digits.prop = max(gs("digits.pval")-2, 2),
      text.NA = ".", backtransf = x$backtransf,
      scientific.pval = gs("scientific.pval"),
      big.mark = gs("big.mark"),
      legend = TRUE,
      ...)
```

Arguments

x	An object of class netmeta or netsplit.
method	A character string indicating which method to split direct and indirect evidence is to be used. Either "Back-calculation" or "SIDDE", can be abbreviated. See Details.
upper	A logical indicating whether treatment comparisons should be selected from the lower or upper triangle of the treatment effect matrices (see list elements TE.fixed and TE.random in the netmeta object).
reference.group	Reference treatment.
baseline.reference	A logical indicating whether results should be expressed as comparisons of other treatments versus the reference treatment or vice versa. This argument is only considered if reference.group is not equal to "".
sep.trts	A character string used in comparison names as separator between treatment labels, e.g., " vs ".
quote.trts	A character used to print around treatment labels.
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.
comb.fixed	A logical indicating whether results for the fixed effects (common effects) network meta-analysis should be printed.
comb.random	A logical indicating whether results for the random effects network meta-analysis should be printed.
show	A character string indicating which comparisons should be printed (see Details).
overall	A logical indicating whether estimates from network meta-analyis should be printed in addition to direct and indirect estimates.
ci	A logical indicating whether confidence intervals should be printed in addition to treatment estimates.
test	A logical indicating whether results of a test comparing direct and indirect estimates should be printed.
digits	Minimal number of significant digits, see print.default.
digits.zval	Minimal number of significant digits for z-value of test of agreement between direct and indirect evidence, see print.default.
digits.pval	Minimal number of significant digits for p-value of test of agreement between direct and indirect evidence, see print.default.
digits.prop	Minimal number of significant digits for direct evidence proportions, see print.default.
text.NA	A character string specifying text printed for missing values.
backtransf	A logical indicating whether printed results should be back transformed. For example, if backtransf=TRUE, results for sm="OR" are printed as odds ratios rather than log odds ratios.

`scientific.pval`
: A logical specifying whether p-values should be printed in scientific notation, e.g., 1.2345e-01 instead of 0.12345.

`big.mark`
: A character used as thousands separator.

`legend`
: A logical indicating whether a legend show be printed.

`warn`
: A logical indicating whether warnings should be printed.

`...`
: Additional arguments (ignored at the moment)

Details

A comparison of direct and indirect treatment estimates can serve as check for consistency of network meta-analysis (Dias et al., 2010).

This function provides two methods to derive indirect estimates:

- Back-calculation method deriving indirect estimates from direct pairwise comparisons and network estimates. The direct evidence proportion as described in König et al. (2013) is used in the calculation of the indirect evidence;
- Separate Indirect from Direct Design Evidence (SIDDE) as described in Efthimiou et al. (2018).

Note, for the back-calculation method, indirect treatment estimates are already calculated in `netmeta` and this function combines and prints these estimates in a user-friendly way. Furthermore, this method is not available for the Mantel-Haenszel and non-central hypergeometric distribution approach implemented in `netmetabin`.

Argument `show` determines which comparisons are printed:

"all"	All comparisons
"both"	Only comparisons contributing both direct and indirect evidence
"with.direct"	Comparisons providing direct evidence
"direct.only"	Comparisons providing only direct evidence
"indirect.only"	Comparisons providing only indirect evidence

Value

An object of class `netsplit` with corresponding `print` and `forest` functions. The object is a list containing the following components:

`comb.fixed, comb.random`
: As defined above.

`comparison`
: A vector with treatment comparisons.

`prop.fixed, prop.random`
: A vector with direct evidence proportions (fixed effect / random effects model).

`fixed, random`
: Results of network meta-analysis (fixed effect / random effects model), i.e., data frame with columns comparison, TE, seTE, lower, upper, z, and p.

`direct.fixed, direct.random`
: Network meta-analysis results based on direct evidence (fixed effect / random

effects model), i.e., data frame with columns comparison, TE, seTE, lower, upper, z, and p.

`indirect.fixed, indirect.random`
: Network meta-analysis results based on indirect evidence (fixed effect / random effects model), i.e., data frame with columns comparison, TE, seTE, lower, upper, z, and p.

`compare.fixed, compare.random`
: Comparison of direct and indirect evidence in network meta-analysis (fixed effect / random effects model), i.e., data frame with columns comparison, TE, seTE, lower, upper, z, and p.

`sm`
: A character string indicating underlying summary measure

`level.comb`
: The level used to calculate confidence intervals for pooled estimates.

`version`
: Version of R package netmeta used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>, Gerta Rücker <ruecker@imbi.uni-freiburg.de>, Orestis Efthimiou <oremiou@gmail.com>

References

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

Efthimiou O, Rücker G, Schwarzer G, Higgins J, Egger M, Salanti G (2018): A Mantel-Haenszel model for network meta-analysis of rare events. *Manuscript submitted for publication*.

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

Puhan MA, Schünemann HJ, Murad MH, et al. (2014): A GRADE working group approach for rating the quality of treatment effect estimates from network meta-analysis. *British Medical Journal*, **349**, g5630

See Also

forest.netsplit, netmeta, netmetabin, netmeasures

Examples

```
data(Woods2010)
#
p1 <- pairwise(treatment, event=r, n=N,
               studlab=author, data=Woods2010, sm="OR")
#
net1 <- netmeta(p1)
#
print(netsplit(net1), digits=2)
print(netsplit(net1), digits=2,
      backtransf=FALSE, comb.fixed=FALSE)
```

```
data(Senn2013)
#
net2 <- netmeta(TE, seTE, treat1.long, treat2.long,
                studlab, data=Senn2013)
#
print(netsplit(net2), digits=2)
# Layout of Puhan et al. (2014), Table 1
print(netsplit(net2), digits=2, ci=TRUE, test=FALSE)

## Not run:
data(Dong2013)
p3 <- pairwise(treatment, death, randomized, studlab = id,
               data = Dong2013, sm = "OR")
net3  <- netmetabin(p3)
netsplit(net3)

## End(Not run)
```

`pairwise` *Transform meta-analysis data from two arm-based formats into contrast-based format*

Description

This function transforms data that are given in wide or long arm-based format (e.g. input format for WinBUGS) to a contrast-based format that is needed as input to R function `netmeta`. The function can transform data with binary, continuous, or generic outcomes as well as incidence rates from arm-based to contrast-based format.

Usage

```
pairwise(treat, event, n, mean, sd, TE, seTE, time,
         data=NULL, studlab,
         incr=0.5, allincr=FALSE, addincr=FALSE, allstudies=FALSE,
         warn = FALSE, ...)
```

Arguments

`treat`	A list or vector with treatment information for individual treatment arms (see Details).
`event`	A list or vector with information on number of events for individual treatment arms (see Details).
`n`	A list or vector with information on number of observations for individual treatment arms (see Details).
`mean`	A list or vector with estimated means for individual treatment arms (see Details).
`sd`	A list or vector with information on the standard deviation for individual treatment arms (see Details).

TE	A list or vector with estimated treatment effects for individual treatment arms (see Details).
seTE	A list or vector with standard errors of estimated treatment effect for individual treatment arms (see Details).
time	A list or vector with information on person time at risk for individual treatment arms (see Details).
data	An optional data frame containing the study information.
studlab	A vector with study labels (optional).
incr	A numerical value which is added to each cell frequency for studies with a zero cell count.
allincr	A logical indicating if incr is added to each cell frequency of all studies if at least one study has a zero cell count. If FALSE (default), incr is added only to each cell frequency of studies with a zero cell count.
addincr	A logical indicating if incr is added to each cell frequency of all studies irrespective of zero cell counts.
allstudies	A logical indicating if studies with zero or all events in two treatment arms are to be included in the meta-analysis (applies only if sm is equal to "RR" or "OR").
warn	A logical indicating whether warnings should be printed (e.g., if studies are excluded due to only providing a single treatment arm).
...	Additional arguments passed-through to the functions to calculate effects.

Details

R function netmeta expects data in a **contrast-based format**, where each row corresponds to a comparison of two treatments and contains a measure of the treatment effect comparing two treatments with standard error, labels for the two treatments and an optional study label. In contrast-based format, a three-arm study contributes three rows with treatment comparison and corresponding standard error for pairwise comparison *A* vs *B*, *A* vs *C*, and *B* vs *C* whereas a four-arm study contributes six rows / pairwise comparisons: *A* vs *B*, *A* vs *C*, ..., *C* vs *D*.

Other programs for network meta-analysis in WinBUGS and Stata require data in an *arm-based* format, i.e. treatment estimate for each treatment arm instead of a difference of two treatments. A common **(wide) arm-based format** consists of one data row per study, containing treatment and other necessary information for all study arms. For example, a four-arm study contributes one row with four treatment estimates and corresponding standard errors for treatments *A*, *B*, *C*, and *D*. Another possible arm-based format is a long format where each row corresponds to a single study arm. Accordingly, in the **long arm-based format** a study contributes as many rows as treatments considered in the study.

The pairwise function transforms data given in (wide or long) arm-based format into the contrast-based format which consists of *pairwise* comparisons and is needed as input to the netmeta function.

The pairwise function can transform data with binary outcomes (using the metabin function from R package meta), continuous outcomes (metacont function), incidence rates (metainc function), and generic outcomes (metagen function). Depending on the outcome, the following arguments are mandatory:

- treat, event, n (see metabin)
- treat, n, mean, sd (see metacont)
- treat, event, time (see metainc)
- treat, TE, seTE (see metagen)

Argument treat is mandatory to identify the individual treatments. The other arguments contain outcome specific data. These arguments must be either lists (wide arm-based format, i.e., one row per study) or vectors (long arm-based format, i.e. multiple rows per study) of the same length.

For the wide arm-based format, each list consists of as many vectors of the same length as the multi-arm study with the largest number of treatments. If a single multi-arm study has five arms, five vectors have to be provided for each lists. Two-arm studies have entries with NA for the third and subsequent vectors. Each list entry is a vector with information for each individual study; i.e. the length of this vector corresponds to the total number of studies incorporated in the network meta-analysis. Typically, list elements are part of a data frame (argument data, optional); see Examples. An optional vector with study labels can be provided which can be part of the data frame.

In the long arm-based format, argument studlab is mandatory to identify rows contributing to individual studies.

Additional arguments for meta-analysis functions can be provided using argument '...'. The following is a list of some important arguments:

Argument	Description	R function
sm	Summary measure	metabin, metacont, metainc, metagen
method	Meta-analysis method	metabin, metainc
method.tau	Estimation of between-study variance	metabin, metacont, metainc, metagen
method.smd	Standardised mean difference	metacont

More information on these as well as other arguments is given in the help pages of R functions metabin, metacont, metainc, and metagen, respectively.

The value of pairwise is a data frame with as many rows as there are pairwise comparisons. For each study with *p* treatments, *p*(p-1)/2* contrasts are generated. Each row contains the treatment effect (TE), its standard error (seTE), the treatments compared ((treat1), (treat2)) and the study label ((studlab)). Further columns are added according to type of data.

All variables from the original dataset are also part of the output dataset. If data are provided in the long arm-based format, the value of a variable can differ between treatment arms; for example, the mean age or percentage of women in the treatment arm. In this situation, two variables instead of one variable will be included in the output dataset. The values "1" and "2" are added to the names for these variables, e.g. "mean.age1" and "mean.age2" for the mean age.

In general, any variable names in the original dataset that are identical to the main variable names (i.e., "TE", "seTE", ...) will be renamed to variable names with ending ".orig".

Value

A data frame with the following columns

TE	Treatment estimate comparing treatment 'treat1' and 'treat2'.
seTE	Standard error of treatment estimate.

studlab	Study labels.
treat1	First treatment in comparison.
treat2	Second treatment in comparison.
event1	Number of events for first treatment arm (for metabin and metainc).
event2	Number of events for second treatment arm (for metabin and metainc).
n1	Number of observations for first treatment arm (for metabin and metacont).
n2	Number of observations for second treatment arm (for metabin and metacont).
mean1	Estimated mean for first treatment arm (for metacont).
mean2	Estimated mean for second treatment arm (for metacont).
sd1	Standard deviation for first treatment arm (for metacont).
sd2	Standard deviation for second treatment arm (for metacont).
TE1	Estimated treatment effect for first treatment arm (for metagen).
TE2	Estimated treatment effect for second treatment arm (for metagen).
seTE1	Standard error of estimated treatment effect for first treatment arm (for metagen).
seTE2	Standard error of estimated treatment effect for second treatment arm (for metagen).
time1	Person time at risk for first treatment arm (for metainc).
time2	Person time at risk for second treatment arm (for metainc).

All variables from the original dataset are also part of the output dataset; see Details.

Author(s)

Gerta Rücker<ruecker@imbi.uni-freiburg.de>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

netmeta, metacont, metagen, metabin, metainc, netgraph

Examples

```
# Example using continuous outcomes (internal call of function metacont)
#
data(parkinson)
# Transform data from arm-based format to contrast-based format
p1 <- pairwise(list(Treatment1, Treatment2, Treatment3),
               n=list(n1, n2, n3),
               mean=list(y1, y2, y3),
               sd=list(sd1, sd2, sd3),
               data=parkinson, studlab=Study)
p1

# Conduct network meta-analysis
net1 <- netmeta(p1)
net1
```

```
# Draw network graphs
netgraph(net1, points=TRUE, cex.points=3, cex=1.5,
         thickness="se.fixed")
netgraph(net1, points=TRUE, cex.points=3, cex = 1.5,
         plastic=TRUE, thickness="se.fixed",
         iterate=TRUE)
netgraph(net1, points=TRUE, cex.points=3, cex = 1.5,
         plastic=TRUE, thickness="se.fixed",
         iterate=TRUE, start="eigen")


#
# Example using generic outcomes (internal call of function metagen)
#
# Calculate standard error for means y1, y2, y3
parkinson$se1 <- with(parkinson, sqrt(sd1^2 / n1))
parkinson$se2 <- with(parkinson, sqrt(sd2^2 / n2))
parkinson$se3 <- with(parkinson, sqrt(sd3^2 / n3))
# Transform data from arm-based format to contrast-based format using
# means and standard errors (note, argument 'sm' has to be used to
# specify that argument 'TE' is a mean difference)
p2 <- pairwise(list(Treatment1, Treatment2, Treatment3),
               TE=list(y1, y2, y3),
               seTE=list(se1, se2, se3),
               n=list(n1, n2, n3),
               data=parkinson, studlab=Study,
               sm="MD")
p2

# Compare pairwise objects p1 (based on continuous outcomes) and p2
# (based on generic outcomes)
all.equal(p1[, c("TE", "seTE", "studlab", "treat1", "treat2")],
          p2[, c("TE", "seTE", "studlab", "treat1", "treat2")])

# Same result as network meta-analysis based on continuous outcomes
# (object net1)
## Not run:
  net2 <- netmeta(p2)
  net2

## End(Not run)


#
# Example with binary data
#
data(smokingcessation)
# Transform data from arm-based format to contrast-based format
# (interal call of metabin function). Argument 'sm' has to be used for
# odds ratio as risk ratio (sm="RR") is default of metabin function.
p3 <- pairwise(list(treat1, treat2, treat3),
               list(event1, event2, event3),
```

```
               list(n1, n2, n3),
               data=smokingcessation,
               sm="OR")
p3

# Conduct network meta-analysis
net3 <- netmeta(p3)
net3

#
# Example with incidence rates
#
data(dietaryfat)

# Transform data from arm-based format to contrast-based format
p4 <- pairwise(list(treat1, treat2, treat3),
               list(d1, d2, d3),
               time=list(years1, years2, years3),
               studlab=ID,
               data=dietaryfat)
p4

# Conduct network meta-analysis using incidence rate ratios (sm="IRR").
# Note, the argument 'sm' is not necessary as this is the default in R
# function metainc called internally
net4 <- netmeta(p4, sm="IRR")
summary(net4)

#
# Example with long data format
#
data(Woods2010)

# Transform data from long arm-based format to contrast-based format
# Argument 'sm' has to be used for odds ratio as summary measure; by
# default the risk ratio is used in the metabin function called
# internally.
p5 <- pairwise(treatment, event=r, n=N,
               studlab=author, data=Woods2010, sm="OR")
p5

# Conduct network meta-analysis
net5 <- netmeta(p5)
net5
```

---

parkinson *Network meta-analysis of treatments for Parkinson's disease*

Description

Network meta-analysis comparing the effects of a number of treatments for Parkinson's disease.

The data are the mean lost work-time reduction in patients given dopamine agonists as adjunct therapy in Parkinson's disease. The data are given as sample size, mean and standard deviation in each trial arm. Treatments are placebo, coded 1, and four active drugs coded 2 to 5. These data are used as an example in the supplemental material of Dias et al. (2013).

Usage

```
data(parkinson)
```

Format

A data frame with the following columns:

Study Study label

Treatment1 Treatment 1

y1 Treatment effect arm 1

sd1 Standard deviation arm 1

n1 Sample size arm 1

Treatment2 Treatment 2

y2 Treatment effect arm 2

sd2 Standard deviation arm 2

n2 Sample size arm 2

Treatment3 Treatment 3

y3 Treatment effect arm 3

sd3 Standard deviation arm 3

n3 Sample size arm 3

Source

Dias S, Sutton AJ, Ades AE and Welton NJ (2013): Evidence synthesis for decision making 2: A generalized linear modeling framework for pairwise and network meta-analysis of randomized controlled trials. *Medical Decision Making* **33**, 607–17

See Also

pairwise, metacont, netmeta, netgraph

Examples

```
data(parkinson)

# Transform data from arm-based format to contrast-based format
p1 <- pairwise(list(Treatment1, Treatment2, Treatment3),
               n=list(n1, n2, n3),
               mean=list(y1, y2, y3),
               sd=list(sd1, sd2, sd3),
               data=parkinson, studlab=Study)
```

```
p1

# Conduct network meta-analysis
net1 <- netmeta(p1)
net1

# Draw network graphs
netgraph(net1, points=TRUE, cex.points=3, cex=1.5,
         thickness="se.fixed")
netgraph(net1, points=TRUE, cex.points=3, cex = 1.5,
         plastic=TRUE, thickness="se.fixed",
         iterate=TRUE)
netgraph(net1, points=TRUE, cex.points=3, cex = 1.5,
         plastic=TRUE, thickness="se.fixed",
         iterate=TRUE, start="eigen")
```

`plot.netposet` *Scatter plot or biplot showing partially order of treatment ranks*

Description

This function generates a scatter plot or biplot of P-scores with an overlay describing partial order of treatment ranks.

Usage

```
## S3 method for class 'netposet'
plot(x,
     plottype = "scatter",
     pooled=ifelse(x$comb.random, "random", "fixed"),
     dim = "2d",
     sel.x = 1, sel.y = 2, sel.z = 3,
     cex = 1, col = "black",
     cex.text = cex, col.text = col,
     adj.x = 0, adj.y = 1,
     offset.x = 0.005, offset.y = -0.005,
     pch = NULL, cex.points = cex, col.points = col,
     col.lines = "black", lty.lines = 1, lwd.lines = 1,
     arrows = FALSE,
     length = 0.05,
     grid = TRUE,
     col.grid = "gray", lty.grid = 2, lwd.grid = 1,
     ...)
```

Arguments

`x`	An object of class `netmeta` (mandatory).

plottype	A character string indicating whether a scatter plot or biplot should be produced, either "scatter" or "biplot". Can be abbreviated.
pooled	A character string indicating whether scatter plot should be drawn for fixed effect ("fixed") or random effects model ("random"). Can be abbreviated.
dim	A character string indicating whether a 2- or 3-dimensional plot should be produced, either "2d" or "3d". Can be abbreviated.
sel.x	A numeric specifying number of outcome to use for the x-axis in a scatterplot (argument is not considered for a biplot).
sel.y	A numeric specifying number of outcome to use for the y-axis in a scatterplot (argument is not considered for a biplot).
sel.z	A numeric specifying number of outcome to use for the z-axis in a scatterplot (argument is not considered for a biplot).
cex	The magnification to be used for treatment labels and points.
col	Colour(s) of treatment labels and points.
cex.text	The magnification to be used for treatment labels.
col.text	Colour(s) of treatment labels.
adj.x	Value(s) in [0, 1] to specify adjustment of treatment labels on x-axis (only considered in 2-D plots); see text.
adj.y	Value(s) in [0, 1] to specify adjustment of treatment labels on y-axis (only considered in 2-D plots); see text.
offset.x	Offset(s) of treatment labels on x-axis (only considered in 2-D plots).
offset.y	Offset(s) of treatment labels on y-axis (only considered in 2-D plots).
pch	Plot symbol(s) for points; no points printed if equal to NULL.
cex.points	Magnification(s) to be used for points.
col.points	Colour(s) of points.
col.lines	Line colour.
lty.lines	Line type.
lwd.lines	Line width.
arrows	A logical indicating whether arrows should be printed (only considered in 2-D plots).
length	Length of arrows; see arrows.
grid	A logical indicating whether grid lines should be added to plot.
col.grid	Colour of grid lines.
lty.grid	Line type of grid lines.
lwd.grid	Line width of grid lines.
...	Additional graphical arguments.

Details

By default (arguments `plottype="scatter"` and `dim="2d"`), a scatter plot is created showing P-scores (see `netrank`) for the first two outcomes considered in the generation of a partially ordered set of treatment ranks (using `netposet`). In addition to the P-scores, the partially order of treatment ranks is shown as lines connecting treatments which is analogous to a Hasse diagram. If argument `dim="3d"`), a 3-D scatter plot is generated showing P-scores for the first three outcomes.

To overcome the restriction of two or three dimension, a biplot (Gabriel, 1971) can be generated using argument `plottype="biplot"`. This is essentially a scatter plot using the first two (`dim="2d"`) or three (`dim="3d"`) components in a principal components analysis (using `prcomp`). Note, if only two / three outcomes are considered in a `netposet` object, a 2-D / 3-D scatter plot is generated instead of a biplot as a principal component analysis is superfluous in such a situation.

Arguments `sel.x` and `sel.y` can be used to select different outcomes to show on x- and y-axis in a 2-D scatter plot; argument `sel.z` can be used accordingly in a 3-D scatter plot. These arguments are ignored for a biplot.

Note, in order to generate 3-D plots (argument `dim="3d"`), R package **rgl** is necessary. Note, under macOS the X.Org X Window System must be available (see `https://www.xquartz.org`).

Author(s)

Gerta Rücker <ruecker@imbi.uni-freiburg.de>, Guido Schwarzer <sc@imbi.uni-freiburg.de>

References

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

Gabriel KR (1971): The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, **58**, 453–67

See Also

`netmeta`, `netrank`, `netposet`, `hasse`

Examples

```
# Use depression dataset
#
data(Linde2015)
#
# Define order of treatments
#
trts <- c("TCA", "SSRI", "SNRI", "NRI",
          "Low-dose SARI", "NaSSa", "rMAO-A", "Hypericum",
          "Placebo")
#
# Outcome labels
#
outcomes <- c("Early response", "Early remission")
#
# (1) Early response
```

```
#
p1 <- pairwise(treat = list(treatment1, treatment2, treatment3),
               event = list(resp1, resp2, resp3),
               n = list(n1, n2, n3),
               studlab = id, data = Linde2015, sm = "OR")
#
net1 <- netmeta(p1, comb.fixed = FALSE,
                seq = trts, ref = "Placebo")
#
# (2) Early remission
#
p2 <- pairwise(treat = list(treatment1, treatment2, treatment3),
               event = list(remi1, remi2, remi3),
               n = list(n1, n2, n3),
               studlab = id, data = Linde2015, sm = "OR")
#
net2 <- netmeta(p2, comb.fixed = FALSE,
                seq = trts, ref = "Placebo")
#
# Partial order of treatment rankings
#
po2 <- netposet(netrank(net1, small.values = "bad"),
                netrank(net2, small.values = "bad"),
                outcomes = outcomes)
#
# Scatter plot
#
plot(po2)

## Not run:
#
# Same scatter plot as only two outcomes considered in netposet()
#
plot(po2, "biplot")

#
# Consider three outcomes
#
#
# Outcome labels
#
outcomes <- c("Early response", "Early remission", "Lost to follow-up")
#
# (3) Loss to follow-up
#
p3 <- pairwise(treat = list(treatment1, treatment2, treatment3),
               event = list(loss1, loss2, loss3),
               n = list(n1, n2, n3),
               studlab = id, data = Linde2015, sm = "OR")
#
net3 <- netmeta(p3, comb.fixed = FALSE,
                seq = trts, ref = "Placebo")
#
```

```
# Partial order of treatment rankings (with three outcomes)
#
po3 <- netposet(netrank(net1, small.values = "bad"),
                netrank(net2, small.values = "bad"),
                netrank(net3, small.values = "good"),
                outcomes = outcomes)
#
# Hasse diagram
#
hasse(po3)
#
# Scatter plot
#
plot(po3)
#
# Biplot (reverse limits of y-axis as biplot is upside down)
#
plot(po3, "bi", xlim = c(-1, 1.7), ylim = c(2.5, -2.5))

## End(Not run)
```

`print.decomp.design` *Print method for objects of class decomp.design*

Description

Print and summary method for objects of class `decomp.design`.

Usage

```
## S3 method for class 'decomp.design'
print(x, digits.Q=gs("digits.Q"),
      showall=FALSE,
      digits.pval.Q=gs("digits.pval.Q"),
      digits.tau2=gs("digits.tau2"),
      scientific.pval=gs("scientific.pval"),
      big.mark=gs("big.mark"), ...)
```

Arguments

`x`	An object of class `decomp.design`.
`digits.Q`	Minimal number of significant digits for Q statistics, see `print.default`.
`showall`	A logical indicating whether results should be shown for all designs or only designs contributing to chi-squared statistics (default).
`digits.pval.Q`	Minimal number of significant digits for p-value of heterogeneity tests, see `print.default`.
`digits.tau2`	Minimal number of significant digits for between-study variance, see `print.default`.

scientific.pval
: A logical specifying whether p-values should be printed in scientific notation, e.g., 1.2345e-01 instead of 0.12345.

big.mark
: A character used as thousands separator.

...
: Additional arguments (ignored at the moment).

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>, Ulrike Krahn <ulrike.krahn@bayer.com>

See Also

decomp.design

Examples

```
data(Senn2013)

net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD")
print(decomp.design(net1))
```

---

print.netbind *Print method for objects of class netbind*

Description

Print method for objects of class netbind.

Usage

```
## S3 method for class 'netbind'
print(x,
      comb.fixed = x$comb.fixed,
      comb.random = x$comb.random,
      ...)
```

Arguments

x
: An object of class netbind or summary.netbind.

comb.fixed
: A logical indicating whether results for the fixed effects (common effects) model should be printed.

comb.random
: A logical indicating whether results for the random effects model should be printed.

...
: Additional arguments (ignored).

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

netbind

Examples

```
data(Linde2016)

# Only consider studies including Face-to-face PST (to reduce runtime
# of example)
#
face <- subset(Linde2016, id %in% c(16, 24, 49, 118))

# Standard random effects NMA model (with placebo as reference treatment)
#
net1 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = face, reference.group = "placebo",
                sm = "OR", comb.fixed = FALSE)

# Additive CNMA model with placebo as inactive component and reference
#
nc1 <- netcomb(net1, inactive = "placebo")

# Combine results of standard NMA and CNMA
#
nb1 <- netbind(nc1, net1,
               name = c("Additive CNMA", "Standard NMA"),
               col.study = c("red", "black"),
               col.square = c("red", "black"))

nb1
print(nb1, comb.fixed = TRUE)
```

---

print.netcomb *Print and summary method for objects of class netcomb*

Description

Print and summary method for objects of class netcomb.

Usage

```
## S3 method for class 'netcomb'
print(x,
      comb.fixed = x$comb.fixed,
      comb.random = x$comb.random,
```

```
      backtransf = x$backtransf,
      nchar.trts = x$nchar.trts,
      digits = gs("digits"),
      digits.zval = gs("digits.zval"),
      digits.pval = gs("digits.pval"),
      digits.pval.Q = max(gs("digits.pval.Q"), 2),
      digits.Q = gs("digits.Q"),
      scientific.pval = gs("scientific.pval"),
      big.mark = gs("big.mark"),
      ...)

## S3 method for class 'netcomb'
summary(object,
        comb.fixed = object$comb.fixed,
        comb.random = object$comb.random, ...)

## S3 method for class 'summary.netcomb'
print(x,
      comb.fixed = x$comb.fixed,
      comb.random = x$comb.random,
      backtransf = x$backtransf,
      nchar.trts = x$nchar.trts,
      digits = gs("digits"),
      digits.zval = gs("digits.zval"),
      digits.pval = gs("digits.pval"),
      digits.pval.Q = max(gs("digits.pval.Q"), 2),
      digits.Q = gs("digits.Q"),
      digits.tau2 = gs("digits.tau2"),
      digits.I2 = gs("digits.I2"),
      scientific.pval = gs("scientific.pval"),
      big.mark = gs("big.mark"),
      ...)
```

Arguments

x	An object of class netcomb or summary.netcomb.
object	An object of class netcomb.
comb.fixed	A logical indicating whether results for the fixed effects (common effects) model should be printed.
comb.random	A logical indicating whether results for the random effects model should be printed.
backtransf	A logical indicating whether results should be back transformed in printouts and forest plots. If backtransf=TRUE, results for sm="OR" are presented as odds ratios rather than log odds ratios, for example.
nchar.trts	A numeric defining the minimum number of characters used to create unique treatment names (see Details).
digits	Minimal number of significant digits, see print.default.

digits.zval	Minimal number of significant digits for z- or t-value, see print.default.
digits.pval	Minimal number of significant digits for p-value of overall treatment effect, see print.default.
digits.pval.Q	Minimal number of significant digits for p-value of heterogeneity tests, see print.default.
digits.Q	Minimal number of significant digits for heterogeneity statistics, see print.default.
digits.tau2	Minimal number of significant digits for between-study variance, see print.default.
digits.I2	Minimal number of significant digits for I-squared statistic, see print.default.
scientific.pval	A logical specifying whether p-values should be printed in scientific notation, e.g., 1.2345e-01 instead of 0.12345.
big.mark	A character used as thousands separator.
...	Additional arguments.

Value

A list is returned by the function summary.netcomb with the same elements as a netcomb object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

netcomb, discomb

Examples

```
data(Linde2016)

# Only consider studies including Face-to-face PST (to reduce runtime
# of example)
#
face <- subset(Linde2016, id %in% c(16, 24, 49, 118))

# Conduct random effects network meta-analysis
#
net1 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = face, reference.group = "placebo",
                sm = "OR", comb.fixed = FALSE)

# Additive model for treatment components
#
nc1 <- netcomb(net1)

summary(nc1)
print(summary(nc1), digits = 2, digits.zval = 3)

## Not run:
```

```
# Conduct random effects network meta-analysis
#
net2 <- netmeta(lnOR, selnOR, treat1, treat2, id,
                data = Linde2016, reference.group = "placebo",
                sm = "OR", comb.fixed = FALSE)

# Additive model for treatment components
#
nc2 <- netcomb(net2)

summary(nc2)
print(summary(nc2), digits = 2, digits.zval = 3)

## End(Not run)
```

`print.netmeta` *Print and summary method for objects of class netmeta*

Description

Print and summary method for objects of class `netmeta`.

Usage

```
## S3 method for class 'netmeta'
print(x, sortvar,
      comb.fixed=x$comb.fixed, comb.random=x$comb.random,
      prediction=x$prediction,
      reference.group=x$reference.group,
      baseline.reference=x$baseline.reference,
      all.treatments=x$all.treatments,
      details=TRUE, ma=TRUE,
      backtransf=x$backtransf, nchar.trts=x$nchar.trts,
      digits=gs("digits"), digits.se=gs("digits.se"),
      digits.pval.Q=max(gs("digits.pval.Q"), 2),
      digits.Q=gs("digits.Q"), digits.tau2=gs("digits.tau2"),
      digits.I2=gs("digits.I2"),
      scientific.pval=gs("scientific.pval"),
      big.mark=gs("big.mark"), ...)

## S3 method for class 'netmeta'
summary(object,
        comb.fixed=object$comb.fixed, comb.random=object$comb.random,
        prediction=object$prediction,
        reference.group=object$reference.group,
        baseline.reference=object$baseline.reference,
        all.treatments=object$all.treatments,
        warn=object$warn, ...)
```

```
## S3 method for class 'summary.netmeta'
print(x, comb.fixed=x$comb.fixed, comb.random=x$comb.random,
      prediction=x$prediction,
      reference.group=x$reference.group,
      baseline.reference=x$baseline.reference,
      all.treatments=x$all.treatments,
      backtransf=x$backtransf, nchar.trts=x$nchar.trts,
      header=TRUE,
      digits=gs("digits"),
      digits.pval.Q=max(gs("digits.pval.Q"), 2),
      digits.Q=gs("digits.Q"),
      digits.tau2=gs("digits.tau2"),
      digits.I2=gs("digits.I2"),
      scientific.pval=gs("scientific.pval"),
      big.mark=gs("big.mark"), ...)
```

Arguments

x	An object of class netmeta or summary.netmeta.
object	An object of class netmeta.
sortvar	An optional vector used to sort individual studies (must be of same length as x$TE).
comb.fixed	A logical indicating whether results for the fixed effects (common effects) model should be printed.
comb.random	A logical indicating whether results for the random effects model should be printed.
prediction	A logical indicating whether prediction intervals should be printed.
reference.group	Reference treatment.
baseline.reference	A logical indicating whether results should be expressed as comparisons of other treatments versus the reference treatment (default) or vice versa. This argument is only considered if reference.group has been specified.
all.treatments	A logical or "NULL". If TRUE, matrices with all treatment effects, and confidence limits will be printed.
details	A logical indicating whether further details for individual studies should be printed.
ma	A logical indicating whether summary results of meta-analysis should be printed.
backtransf	A logical indicating whether results should be back transformed in printouts and forest plots. If backtransf=TRUE, results for sm="OR" are presented as odds ratios rather than log odds ratios, for example.
nchar.trts	A numeric defining the minimum number of characters used to create unique treatment names.
header	A logical indicating whether information on title of meta-analysis, comparison and outcome should be printed at the beginning of the printout.

digits	Minimal number of significant digits, see print.default.
digits.se	Minimal number of significant digits for standard deviations and standard errors, see print.default.
digits.pval.Q	Minimal number of significant digits for p-value of heterogeneity tests, see print.default.
digits.Q	Minimal number of significant digits for heterogeneity statistics, see print.default.
digits.tau2	Minimal number of significant digits for between-study variance, see print.default.
digits.I2	Minimal number of significant digits for I-squared statistic, see print.default.
scientific.pval	A logical specifying whether p-values should be printed in scientific notation, e.g., 1.2345e-01 instead of 0.12345.
big.mark	A character used as thousands separator.
warn	A logical indicating whether the use of summary.meta in connection with metacum or metainf should result in a warning.
...	Additional arguments.

Value

A list is returned by the function summary.netmeta with the following elements:

comparison	Results for pairwise comparisons (data frame with columns studlab, treat1, treat2, TE, seTE, lower, upper, z, p).
comparison.nma.fixed	Results for pairwise comparisons based on fixed effect model (data frame with columns studlab, treat1, treat2, TE, seTE, lower, upper, z, p, leverage).
comparison.nma.random	Results for pairwise comparisons based on random effects model (data frame with columns studlab, treat1, treat2, TE, seTE, lower, upper, z, p).
fixed	Results for fixed effect model (a list with elements TE, seTE, lower, upper, z, p).
random	Results for random effects model (a list with elements TE, seTE, lower, upper, z, p).
predict	Prediction intervals (a list with elements seTE, lower, upper).
studies	Study labels coerced into a factor with its levels sorted alphabetically.
narms	Number of arms for each study.
k	Total number of studies.
m	Total number of pairwise comparisons.
n	Total number of treatments.
d	Total number of designs (corresponding to the unique set of treatments compared within studies).
Q	Overall heterogeneity / inconsistency statistic.
df.Q	Degrees of freedom for test of heterogeneity / inconsistency.
pval.Q	P-value for test of heterogeneity / inconsistency.

`I2` I-squared.

`tau` Square-root of between-study variance.

`Q.heterogeneity`
Overall heterogeneity statistic.

`df.Q.heterogeneity`
Degrees of freedom for test of overall heterogeneity.

`pval.Q.heterogeneity`
P-value for test of overall heterogeneity.

`Q.inconsistency`
Overall inconsistency statistic.

`df.Q.inconsistency`
Degrees of freedom for test of overall inconsistency.

`pval.Q.inconsistency`
P-value for test of overall inconsistency.

`sm` A character string indicating underlying summary measure.

`method` A character string indicating which method is to be used for pooling of studies.

`level` The level used to calculate confidence intervals for individual studies.

`level.comb` The level used to calculate confidence intervals for pooled estimates.

`comb.fixed, comb.random`
As defined above.

`prediction, level.predict`
As defined above.

`reference.group, baseline.reference`
As defined above.

`all.treatments, backtransf`
As defined above.

`ci.lab` Label for confidence interval.

`seq` A character specifying the sequence of treatments.

`tau.preset` An optional value for the square-root of the between-study variance τ^2.

`sep.trts` A character used in comparison names as separator between treatment labels.

`nchar.trts` A numeric defining the minimum number of characters used to create unique treatment names.

`title` Title of meta-analysis / systematic review.

`call` Function call.

`version` Version of R package netmeta used to create object.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

netmeta

Examples

```
data(Senn2013)

#
# Conduct fixed effects network meta-analysis
#
net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD",
                comb.random=FALSE)
print(net1, ref="plac", digits=3)
summary(net1)

#
# Conduct random effects network meta-analysis
#
net2 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD",
                comb.fixed=FALSE)
print(net2, ref="plac", digits=3)
summary(net2)
```

Senn2013 *Network meta-analysis in diabetes*

Description

Network meta-analysis in diabetes comparing effects of a number of drugs on the HbA1c value.

These data are used as an example in Senn et al. (2013) and have been preprocessed for use in R package netmeta.

Usage

```
data(Senn2013)
```

Format

A data frame with the following columns:

TE Treatment effect

seTE Standard error of treatment effect

treat1 Treatment 1

treat2 Treatment 2

treat1.long Treatment 1 (full treatment names)

treat2.long Treatment 2 (full treatment names)

studlab Study label

Details

Treatment labels provided by columns `treat1` and `treat2` have been abbreviated:

- acar = Acarbose
- benf = Benfluorex
- metf = Metformin
- migl = Miglitol
- piog = Pioglitazone
- plac = Placebo
- rosi = Rosiglitazone
- sita = Sitagliptin
- sulf = Sulfonylurea
- vild = Vildagliptin

Full treatment names are available in columns `treat1.long` and `treat2.long`.

Source

Senn S, Gavini F, Magrez D, Scheen A (2013): Issues in performing a network meta-analysis. *Statistical Methods in Medical Research*, **22**, 169–89

See Also

netmeta

Examples

```
data(Senn2013)

#
# Fixed effect model
#
net1 <- netmeta(TE, seTE, treat1, treat2,
                studlab, data=Senn2013,
                comb.random = FALSE)
net1
net1$Q.decomp

#
# Forest plot
#
forest(net1, ref="plac")

## Not run:
#
# Comparison with reference group
#
netmeta(TE, seTE, treat1.long, treat2.long,
```

```
        studlab, data=Senn2013,
        reference="plac")

#
# Random effects model
#
net2 <- netmeta(TE, seTE, treat1, treat2, studlab, data=Senn2013,
                comb.fixed = FALSE)
net2
forest(net2, ref="plac")

## End(Not run)
```

`smokingcessation` *Network meta-analysis of interventions for smoking cessation*

Description

Network meta-analysis comparing the effects of a number of interventions for smoking cessation.

These data are used as an example in Dias et al. (2013), page 651.

Usage

```
data(smokingcessation)
```

Format

A data frame with the following columns:

event1 Number of individuals with successful smoking cessation in arm 1

n1 Number of individuals in arm 1

event2 Number of individuals with successful smoking cessation in arm 2

n2 Number of individuals in arm 2

event3 Number of individuals with successful smoking cessation in arm 3

n3 Number of individuals in arm 3

treat1 Treatment 1

treat2 Treatment 2

treat3 Treatment 3

Source

Dias S, Welton NJ, Sutton AJ, Caldwell DM, Lu G and Ades AE (2013): Evidence Synthesis for Decision Making 4: Inconsistency in networks of evidence based on randomized controlled trials. *Medical Decision Making*, **33**, 641–56

See Also

pairwise, metabin, netmeta, netgraph

Examples

```
data(smokingcessation)

# Transform data from arm-based format to contrast-based format
# Argument 'sm' has to be used for odds ratio as summary measure; by
# default the risk ratio is used in the metabin function called
# internally.
p1 <- pairwise(list(treat1, treat2, treat3),
               event=list(event1, event2, event3),
               n=list(n1, n2, n3),
               data=smokingcessation,
               sm="OR")
p1

# Conduct network meta-analysis
net1 <- netmeta(p1)
net1

# Draw network graph
netgraph(net1, points=TRUE, cex.points=3, cex=1.25)
tname <- c("No intervention", "Self-help",
           "Individual counselling", "Group counselling")
netgraph(net1, points=TRUE, cex.points=3, cex=1.25, labels=tname)
```

---

treats *Abbreviate treatment names*

Description

Auxiliary function to create uniquely abbreviated treatment names.

Usage

```
treats(x, nchar.trts = 8, row = TRUE)
```

Arguments

x	A vector with treatment names or a matrix with treatment names as row and / or column names.
nchar.trts	A numeric defining the minimum number of characters used to create unique treatment names.
row	A logical indicating whether row or column names should be used (only considered if argument x is a matrix).

Details

This auxiliary function can be used to create uniquely abbreviated treatment names (and is used internally in several R functions for this purpose).

Initially, to construct uniquely abbreviated treatment names, `substring` is used to extract the first `nchar.trts` characters. If these abbreviated treatment names are not unique, `abbreviate` with argument `minlength=nchar.trts` is used.

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>

See Also

`netmeta`, `print.netmeta`, `print.summary.netmeta`

Examples

```
data(Senn2013)
#
net1 <- netmeta(TE, seTE, treat1, treat2,
                studlab, data=Senn2013)

# Use matrix with fixed effect treatment estimates to create unique
# treatment names
#
# Four characters
#
treats(net1$TE.fixed, nchar.trts = 4)

# Two characters
#
treats(net1$TE.fixed, nchar.trts = 2)

# One character
#
treats(net1$TE.fixed, nchar.trts = 1)
```

Woods2010 *Count statistics of survival data*

Description

Count mortality statistics in randomised controlled trials of treatments for chronic obstructive pulmonary disease (Woods et al. (2010), Table 1).

Usage

```
data(Woods2010)
```

Format

A data frame with the following columns:

author First author / study name

treatment Treatment

r Number of deaths in treatment arm

N Number of patients in treatment arm

Source

Woods BS, Hawkins N, Scott DA (2010): Network meta-analysis on the log-hazard scale, combining count and hazard ratio statistics accounting for multi-arm trials: A tutorial. *BMC Medical Research Methodology*, **10**, 54

See Also

pairwise, metabin, netmeta

Examples

```
data(Woods2010)

# Transform data from long arm-based format to contrast-based format
# Argument 'sm' has to be used for odds ratio as summary measure; by
# default the risk ratio is used in the metabin function called
# internally.
p1 <- pairwise(treatment, event = r, n = N,
               studlab = author, data = Woods2010, sm = "OR")
p1

# Conduct network meta-analysis
net1 <- netmeta(p1)
net1

## Not run:
# Show forest plot
forest(net1, ref = "Placebo", drop = TRUE,
       leftlabs = "Contrast to Placebo")

## End(Not run)
```

Index