Package 'netmeta'

July 29, 2014

Title Network meta-analysis with R

Version 0.6-0

Depends meta

Suggests colorspace

Imports magic

Date 2014-07-29

Description Network meta-analysis following methods by Rücker (2012) and Krahn et al. (2013)

License GPL (>= 2)

Encoding UTF-8

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

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

NeedsCompilation no

Repository CRAN

Date/Publication 2014-07-29 18:14:49

R topics documented:

`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)

##
## Fixed effect model (default)
##
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)
```

Arguments

`x`	An object of class `netmeta`.
`tau.preset`	An optional value for the square-root of the between-study variance tau^2 of a random effects model on which a between-designs Q statistic (see `Q.inc.random.preset`) will be based. Design-specific contributions of this Q statistic (see `Q.inc.design.random.preset`) as well as residuals after detaching of single designs (see `residuals.inc.detach.random.preset`) can be obtained.

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 <krahnu@uni-mainz.de>, 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**(2), 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**(29), 3805–3820.

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)
```

---

forest.netmeta *Forest plot*

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,
       leftcols="studlab",
       leftlabs="Treatment",
       smlab=NULL,
       sortvar=x$seq,
       ...)
```

Arguments

x	An object of class netmeta.
reference.group	Reference group.
pooled	A character string indicating whether results for 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).
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).
...	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. 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.

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)

##
## Fixed effect model (default)
##
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",
```

```
       leftcols="studlab", rightcols=NULL,
       leftlabs="Contrast to placebo")

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

forest(net2, xlim=c(-1.5,1), ref="plac",
       xlab="HbA1c difference",
       leftcols="studlab", rightcols=NULL,
       leftlabs="Contrast to placebo")
```

---

netgraph *Network graph*

Description

This function generates a graph of the evidence network.

Usage

```
netgraph(x, seq=x$seq,
         labels=dimnames(x$A.matrix)[[1]],
         cex=1, col="slateblue", offset=0.0175, scale=1.10,
         plastic, thickness, lwd=5, lwd.min=lwd/2.5, lwd.max=lwd*4,
         highlight=NULL, col.highlight="red2",
         lwd.highlight=lwd, highlight.split=":",
         multiarm=any(x$narms>2), col.multiarm=NULL,
         alpha.transparency=0.5,
         points=FALSE, col.points="red", cex.points=1, pch.points=20,
         start.layout="circle", eig1=2, eig2=3,
         iterate, tol=0.0001, maxit=500, allfigures=FALSE,
         A.matrix=x$A.matrix, N.matrix=sign(A.matrix),
         xpos=NULL, ypos=NULL,
         ...)
```

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	Color of lines connecting treatments if argument plastic=FALSE.

`offset` Distance between edges (i.e. treatments) in graph and treatment labels (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.

`thickness` Either a character variable to determine the method to plot line widths (see Details) or a matrix of the same dimension 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`.

`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`.

`highlight.split`
A character defining splitting criterion (only necessary if colon is used in treatment labels).

`multiarm` A logical indicating whether multi-arm studies should 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 edges (i.e. treatments) of the network graph.

`col.points, cex.points, pch.points`
Corresponding color, size, type for points.

`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.

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 (*n*x*n*) 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 (*n*x*n*) 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.
xpos	Vector (*n*) of x coordinates.
ypos	Vector (*n*) of y coordinates.
...	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". 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).

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.fixed")
- 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.

Author(s)

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

References

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

Hall, K.M. (1970). An r-dimensional quadratic placement algorithm. *Management Science*, **17**, 219–229.

Hu, Y. (2012). *Combinatorial Scientific Computing*, Chapter Algorithms for Visualizing Large Networks, pages 525–549. Chapman and Hall/CRC Computational Science.

Kamada, T. and Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information Processing Letters*, **31**(1), 7–15.

Michailidis, G. and de Leeuw, J. (2001). Data visualization through graph drawing. *Computational Statistics*, **16**(3), 435–450.

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="acar:metf", 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="acar:metf")

## 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
##
netgraph(net1, start="circle", iterate=TRUE, col.multiarm="purple",
         allfigures=TRUE)

## 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 3D look of the comparisons and one
## highlighted comparison
##
netgraph(net1, plastic=FALSE, highlight="acar:metf")

## Network graph without 3D 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])
```

---

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, ...)
```

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.
...	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.

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 <krahnu@uni-mainz.de>

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**(29), 3805–3820.

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)
```

---

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)
```

Arguments

x	An object of class netmeta.

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:

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 <krahnu@uni-mainz.de>, 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**(30), 5414–29.

See Also

netmeta

Examples

```
data(Senn2013)

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

##
## 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)

## Generation of an object of class 'netmeta' with
## reference treatment 'plac' based on a random
## effects model
##
net2 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD", reference="plac", comb.random=TRUE)

##
## 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=0.95, level.comb=0.95,
        comb.fixed=TRUE, comb.random=FALSE, reference.group="",
        all.treatments=NULL, seq=NULL, tau.preset=NULL, title="", warn=TRUE)
```

Arguments

TE	Estimate of treatment effect, i.e. difference between first and second treatment (e.g. log odds ratio or mean difference).
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", "AS", "MD", "SMD".
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 effect meta-analysis should be conducted.
comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
reference.group	Reference group.
all.treatments	A logical or value "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 the square-root of the between-study variance τ^2.
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

Let *n* be the number of different treatments 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 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).

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.

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

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 (after appropriate adjustment for multi-arm studies). Then, as in standard pairwise meta-analysis, the procedure is repeated with the resulting enlarged standard errors.

Value

An object of class `c("netmeta")` with corresponding `print`, `summary`, `forest` function. The object is a list containing the following components:

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

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

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

`reference.group`
: The name of the reference group, if specified, otherwise `c("")`.

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

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

`narms` Number of arms for each 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).

`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).

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

`k` Total number of studies.

`m` Total number of pairwise comparisons.

`n` Total number of treatments.

`Q` Overall heterogeneity / inconsistency statistic.

`df` 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.

`Q.inconsistency`
: Overall inconsistency statistic.

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

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

`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 and Riley RD (2012), Quantifying the impact of between-study heterogeneity in multivariate meta-analyses. *Statistics in Medicine*, **31**(29). 3805–3820.

Rücker G (2012), Network meta-analysis, electrical networks and graph theory. *Research Synthesis Methods*, **3**, 312–324.

Senn S, Gavini F, Magrez D, and Scheen A (2013). Issues in performing a network meta-analysis. *Statistical Methods in Medical Research*, **22**(2), 169–189. First published online 2012 Jan 3.

See Also

`forest.netmeta`, `metagen`

Examples

```
data(Senn2013)

##
## Fixed effect model (default)
##
net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD")
net1
net1$Q.decomp
```

```
##
## Comparison with reference group
##
netmeta(TE, seTE, treat1, treat2, studlab,
        data=Senn2013, sm="MD", reference="plac")

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

##
## Change printing order of treatments (placebo first)
##
trts <- c("plac", "acar", "benf", "metf", "migl", "piog",
          "rosi", "sita", "sulf", "vild")
net3 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD",
                seq=trts)
print(summary(net3), digits=2)
```

`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=2, ...)
```

Arguments

`x`	An object of class `decomp.design`.
`digits`	Minimal number of significant digits for Q statistics, see `print.default`.
`...`	Additional arguments (ignored at the moment).

Author(s)

Guido Schwarzer <sc@imbi.uni-freiburg.de>, Ulrike Krahn <krahnu@uni-mainz.de>

See Also

`decomp.design`

Examples

```
data(Senn2013)

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

`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, level=x$level, level.comb=x$level.comb,
      comb.fixed=x$comb.fixed, comb.random=x$comb.random,
      reference.group=x$reference.group, all.treatments=x$all.treatments,
      details=TRUE, ma=TRUE, logscale=FALSE,
      digits=max(4, .Options$digits - 3), ...)

## S3 method for class 'netmeta'
summary(object,
        level=object$level, level.comb=object$level.comb,
        comb.fixed=object$comb.fixed, comb.random=object$comb.random,
       reference.group=object$reference.group, all.treatments=object$all.treatments,
        warn=object$warn, ...)

## S3 method for class 'summary.netmeta'
print(x, comb.fixed=x$comb.fixed, comb.random=x$comb.random,
      reference.group=x$reference.group, all.treatments=x$all.treatments,
      logscale=FALSE, header=TRUE, digits=max(3, .Options$digits - 3), ...)
```

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`).
`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 effect meta-analysis should be conducted.

comb.random	A logical indicating whether a random effects meta-analysis should be conducted.
reference.group	Reference group.
all.treatments	A logical or value "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.
logscale	A logical indicating whether results for summary measures 'RR', 'OR', 'HR', or 'PLN' will be printed on logarithmic scale.
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.
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 (a list with elements TE, seTE, lower, upper, z, p, level, df, studlab, treat1, treat2).
comparison.nma.fixed	Results for pairwise comparisons based on fixed effect model (a list with elements TE, seTE, lower, upper, z, p, level, df, studlab, treat1, treat2, leverage).
comparison.nma.random	Results for pairwise comparisons based on random effects model (a list with elements TE, seTE, lower, upper, z, p, level, df, studlab, treat1, treat2).
fixed	Results for fixed effect model (a list with elements TE, seTE, lower, upper, z, p, level, df).
random	Results for random effects model (a list with elements TE, seTE, lower, upper, z, p, level, df).
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.
Q	Overall heterogeneity / inconsistency statistic.
df	Degrees of freedom for test of heterogeneity / inconsistency.
tau	Square-root of between-study variance.
I2	I-squared.

sm	A character string indicating underlying summary measure.
ci.lab	Label for confidence interval.
comb.fixed	A logical indicating whether result for fixed effect meta-analysis should be printed.
comb.random	A logical indicating whether result for random effects meta-analysis should be printed.
level	The level used to calculate confidence intervals for individual comparisons.
level.comb	The level used to calculate confidence intervals for pooled estimates.
seq	A character specifying the sequence of treatments.
all.treatments	A logical or value "NULL". If TRUE, matrices with all treatment effects, and confidence limits will be printed.
reference.group	Reference group.
all.treatments	A logical or value "NULL". If TRUE, matrices with all treatment effects, and confidence limits will be printed.
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)

##
## Fixed effect model (default)
##
net1 <- netmeta(TE, seTE, treat1, treat2, studlab,
                data=Senn2013, sm="MD")
print(net1, ref="plac", digits=3)
summary(net1)

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

studlab Study label

Details

Treatment labels have been abbreviated:

- acar = Acarbose
- benf = Benfluorex
- metf = Metformin
- migl = Miglitol
- piog = Pioglitazone
- plac = Placebo
- rosi = Rosiglitazone
- sita = Sitagliptin
- sulf = Sulfonylurea
- vild = Vildagliptin

Source

Senn S, Gavini F, Magrez D, and Scheen A (2013). Issues in performing a network meta-analysis. *Statistical Methods in Medical Research*, **22**(2), 169–189. First published online 2012 Jan 3.

See Also

netmeta

Examples

```
data(Senn2013)

##
## Fixed effect model (default)
##
net1 <- netmeta(TE, seTE, treat1, treat2, studlab, data=Senn2013)
net1
net1$Q.decomp


##
## Comparison with reference group
##
netmeta(TE, seTE, treat1, treat2, studlab, data=Senn2013,
        reference="plac")

##
## Forest plot
##
forest(net1, ref="plac")

##
## Random effects model
##
net2 <- netmeta(TE, seTE, treat1, treat2, studlab, data=Senn2013,
                comb.random = TRUE)
net2
forest(net2, ref="plac")
```

Index