Package ‘crossnma’

October 12, 2022

Title Cross-Design & Cross-Format Network Meta-Analysis and Regression

Version 1.0.1

Date 2022-04-13

Depends R (>= 3.5), netmeta

Imports rjags, coda, dplyr, plyr, rlang, magrittr, tidyr, ggplot2

Suggests rmarkdown, knitr

URL `https://github.com/htx-r/crossnma`

Description Network meta-analysis and meta-regression (allows including up to 3 covariates) for individual participant data, aggregate data, and mixtures of both formats using the three-level hierarchical model. Each format can come from randomized controlled trials or non-randomized studies or mixtures of both. Estimates are generated in a Bayesian framework using JAGS. The implemented models are described by Hamza et al. 2022 <DOI:10.48550/arXiv.2203.06350>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.2

NeedsCompilation no

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

Author Tasnim Hamza [aut] (<`https://orcid.org/0000-0002-4700-6990`>),
Guido Schwarzer [aut, cre] (<`https://orcid.org/0000-0001-6214-9087`>),
Georgia Salanti [aut] (<`https://orcid.org/0000-0002-3830-8508`>)

Repository CRAN

Date/Publication 2022-04-15 10:02:29 UTC

R topics documented:

`crossnma-package` *crossnma: An R package for synthesizing cross-design evidence and cross-format data using Bayesian methods in network meta-analysis and network meta-regression*

Description

An R package **crossnma** for performing (network) meta-analysis and (network) meta-regression (allows including up to 3 covariates) of individual participant data and aggregate data or combination of both. Each format can come from randomized controlled trials or non-randomized studies. Estimates are generated in a Bayesian framework using JAGS. The implemented models are described by Hamza et al. 2022 doi: 10.48550/arXiv.2203.06350.

Details

The evidence in network meta-analysis (NMA) typically comes from randomized controlled trials (RCT) where aggregate data (AD) are extracted from published reports. Retrieving individual participant data (IPD) allows considering participant covariates to explain some of the heterogeneity/inconsistency in the network and identify effect modifiers. Additionally, evidence from non-randomized studies (NRS) reflects the reality in clinical practice and bridges the efficacy-effectiveness gap. The cross-NMA/NMR model is a Bayesian suite for evidence synthesis which extends and integrates four different approaches that combine RCT and NRS evidence into a three-level hierarchical model for the synthesis of IPD and AD. The four approaches account for differences in the design and risk of bias in the RCT and NRS evidence. These four approaches variously ignoring differences in risk of bias, using NRS to construct penalized treatment effect priors and bias-adjustment models that control the contribution of information from high risk of bias studies in two different ways.

Further details:

- To have a list of all R functions available in **crossnma** type `help(package = "crossnma")`
- The R command `citation("crossnma")` shows how to cite **crossnma** in publications.

- To report problems and bugs send an email to <`tasnim.hamza@ispm.unibe.ch`>
- The development version of **crossnma** is available on GitHub `https://github.com/htx-r/crossnma`.

Author(s)

Tasnim Hamza <`tasnim.hamza@ispm.unibe.ch`>, Guido Schwarzer <`sc@imbi.uni-freiburg.de`>, Georgia Salanti <`georgia.salanti@ispm.unibe.ch`>

References

Saramago P, Sutton AJ, Cooper NJ, Manca A (2012): Mixed treatment comparisons using aggregate and individual participant level data. *Statistics in Medicine*, **10;31(28)**, 3516-36

Dias, Sofia, N. J. Welton, V. C. C. Marinho, G. Salanti, J.P.T Higgins, and A. E. Ades (2010): Estimation and Adjustment of Bias in Randomized Evidence by Using Mixed Treatment Comparison Meta-Analysis. *Journal of the Royal Statistical Society*, **173**, 613-29

Plummer, Martyn. (2003): JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling.

Tramacere, Del Giovane, I, and G Filippini (2015): Immunomodulators and Immunosuppressants for Relapsing-remitting Multiple Sclerosis: A Network Meta-analysis. *Cochrane Database of Systematic Reviews, no. 9.* John Wiley & Sons, Ltd. doi: 10.1002/14651858.CD011381.pub2.

Verde, Pablo Emilio. (2020): A Bias-Corrected Meta-Analysis Model for Combining, Studies of Different Types and Quality. *Biometrical Journal*, doi: 10.1002/bimj.201900376

`crossnma` *Run JAGS to fit cross NMA and NMR*

Description

This function takes the JAGS model from an object produced by `crossnma.model` and runs it using `jags.model` in rjags package.

Usage

```
crossnma(
  x,
  n.adapt = 1000,
  n.burnin = floor(n.iter/2),
  n.iter = 10000,
  thin = 1,
  n.chains = 2,
  quiet = TRUE,
  monitor = NULL
)
```

Arguments

`x`	An object produced by `crossnma.model`.
`n.adapt`	Number of adaptations for the MCMC chains. Default is 1000.
`n.burnin`	Number of burnin iterations for the MCMC chains.
`n.iter`	Number of iterations for the MCMC chains.
`thin`	Number of thinning for the MCMC chains. Default is 1.
`n.chains`	Number of MCMC chains. Default is 2.
`quiet`	A logical passed on to `jags.model`.
`monitor`	A vector of additional parameters to monitor. Default is NULL.

Value

An object of class `crossnma` which is a list containing the following components:

`samples`	The MCMC samples produced by running the JAGS model.
`model`	The `crossnma.model` object obtained from `crossnma.model` which was used to run JAGS.
`trt.key`	A table of treatment names and their correspondence to integers used in the JAGS model.
`call`	Function call.
`version`	Version of R package **crossnma** used to create object.

Author(s)

Tasnim Hamza <`tasnim.hamza@ispm.unibe.ch`>, Guido Schwarzer <`sc@imbi.uni-freiburg.de`>

See Also

`crossnma.model`, `jags.model`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")

# Fit JAGS model
# (suppress warning 'Adaptation incomplete' due to n.adapt = 20)
fit <-
```

```
  suppressWarnings(crossnma(mod, n.adapt = 20,
    n.iter = 50, thin = 1, n.chains = 3))

# Display the output
summary(fit)
plot(fit)
```

`crossnma.model` *Create JAGS model and data to perform cross network meta-analysis or meta-regression*

Description

This function creates a JAGS model and the needed data for cross-design and cross-format network meta-analysis or meta-regression of a binary outcome with the odds ratio as effect measure.

Usage

```
crossnma.model(
  trt,
  study,
  outcome,
  n,
  design,
  cov1 = NULL,
  cov2 = NULL,
  cov3 = NULL,
  bias = NULL,
  unfav = NULL,
  bias.covariate = NULL,
  bias.group = NULL,
  prt.data = NULL,
  std.data = NULL,
  reference = NULL,
  trt.effect = "random",
  cov1.ref = NULL,
  cov2.ref = NULL,
  cov3.ref = NULL,
  reg0.effect = "independent",
  regb.effect = "random",
  regw.effect = "random",
  split.regcoef = TRUE,
  method.bias = NULL,
  bias.type = NULL,
  bias.effect = "common",
  down.wgt = NULL,
```

```
  prior = list(tau.trt = NULL, tau.reg0 = NULL, tau.regb = NULL, tau.regw = NULL,
    tau.gamma = NULL, pi.high.rct = NULL, pi.low.rct = NULL, pi.high.nrs = NULL,
    pi.low.nrs = NULL),
  run.nrs = list(var.infl = 1, mean.shift = 0, trt.effect = "common", n.adapt = 2000,
    n.iter = 10000, n.burnin = 4000, thin = 1, n.chains = 2),
  quiet = TRUE
)
```

Arguments

`trt`	Treatment variable in prt.data and std.data.
`study`	Study variable in prt.data and std.data.
`outcome`	Outcome variable in prt.data and std.data.
`n`	Number of participants in std.data.
`design`	Design variable in prt.data and std.data.
`cov1`	Optional first covariate in prt.data and std.data to conduct network meta-regression (see Details).
`cov2`	Optional second covariate in prt.data and std.data to conduct network meta-regression (see Details).
`cov3`	Optional third covariate in prt.data and std.data to conduct network meta-regression (see Details).
`bias`	Variable with information on risk of bias in prt.data and std.data (can be provided when method.bias = 'adjust1' or 'adjust2'). Possible values of this variable are 'low', 'high' or 'unclear' (can be abbreviated). These values must be identical for all participants from the same study.
`unfav`	Variable in prt.data and std.data indicating the unfavored treatment in each study (can be provided when method.bias = 'adjust1' or 'adjust2'). The entries of this variable are either 0 (unfavored treatment) or 1 (favorable treatment or treatments). Each study should include only one 0 entry. The values need to be repeated for participants who take the same treatment.
`bias.covariate`	Variable in prt.data and std.data used to estimate the probability of bias (can be provided when method.bias = 'adjust1' or 'adjust2').
`bias.group`	An optional variable in prt.data and std.data that indicates the bias effect in each study (can be provided when method.bias = 'adjust1' or 'adjust2'). The entries of these variables should be either 1 (study has inactive treatment and its estimate should be adjusted for bias effect), 2 (study has only active treatments and its estimate should be adjusted for bias effect (different from inactive bias effect) or 0 (study does not need any bias adjustment). The values need to be repeated for the participants assigned to the same treatment. Default is 1.
`prt.data`	An object of class data.frame containing the individual participant dataset. Each row contains the data of a single participant. The dataset needs to have the following columns: treatment, study identification, outcome (event and non-event), design. Additional columns might be required for certain analyses.

`std.data`	An object of class data.frame containing the study-level dataset. Each row represents the information of study arm. The dataset needs to have the following columns: treatment, study identification, outcome (number of events), sample size and design. Additional columns might be required for certain analyses.
`reference`	A character indicating the name of the reference treatment. When the reference is not specified, the first alphabetic treatment will be used as a reference in the analysis.
`trt.effect`	A character defining the model for the study-specific treatment effects. Options are 'random' (default) or 'common'.
`cov1.ref`	An optional value to center the first covariate which is only useful for a continuous covariate. Dichotomous covariates should be given NA value. The default is the overall minimum covariate value from all studies.
`cov2.ref`	An optional value to center the second covariate which is only useful for a continuous covariate. Dichotomous covariates should be given NA value. The default is the overall minimum covariate value from all studies.
`cov3.ref`	An optional value to center the third covariate which is only useful for a continuous covariate. Dichotomous covariates should be given NA value. The default is the overall minimum covariate value from all studies.
`reg0.effect`	An optional character (needed when at least `cov1` is not NULL) indicating the relationship across studies for the prognostic effects expressed by the regression coefficient, (β_0), in a study j. Options are 'independent' or 'random'. We recommend using 'independent' (default).
`regb.effect`	An optional character (needed when at least `cov1` is not NULL) indicating the relationship across treatments for the between-study regression coefficient (β^B). This parameter quantifies the treatment-mean covariate interaction. Options are 'independent', 'random' or 'common'. Default is 'random'.
`regw.effect`	An optional character (needed when at least `cov1` is not NULL) indicating the relationship across treatments for the within-study regression coefficient (β^W). This parameter quantifies the treatment-covariate interaction effect at the individual level. Options are 'independent', 'random' and 'common'. Default is 'random'.
`split.regcoef`	A logical value (needed when at least `cov1` is not NULL). If TRUE (default) the within- and between-study coefficients will be splitted in the analysis of prt.data. When the split.regcoef = FALSE, only a single regression coefficient will be estimated to represent both the between-studies and within-studies covariate effects. In this case, both arguments `regb.effect` and `regw.effect` need to be given the same option to model the single regression effect.
`method.bias`	A character for defining the method to combine randomized clinical trials (RCT) and non-randomized studies (NRS). Options are 'naive' for naive or unadjusted synthesize, 'prior' for using NRS evidence to construct priors for the relative treatment effects in RCTs analysis, or 'adjust1' and 'adjust2' to allow a bias adjustment. When only one design is available (either rct or nrs), this argument needs also to be specified to indicate whether unadjusted (naive) or bias-adjusted analysis (adjust1 or adjust2) should be applied.
`bias.type`	An optional character defining of bias on the treatment effect (required when method.bias='adjust1'). Three options are possible: 'add' to add the additive

	bias effect,’mult’ for multiplicative bias effect and ’both’ includes both an additive and a multiplicative terms.
`bias.effect`	An optional character indicating the relationship for the bias coefficients across studies. Options are ’random’ or ’common’ (default). It is required when method.bias=’adjust1’ or ’adjust2’.
`down.wgt`	An optional numeric indicating the percent to which studies at high risk of bias will be downweighed on average. The value ranges between 0 and 1. It can be provided when method.bias=’adjust1’ or ’adjust2’.
`prior`	An optional list to control the prior for various parameters in JAGS model. When effects are set as ’random’, we can set the heterogeneity parameters for: tau.trt for the treatment effects, tau.reg0 for the effect of prognostic covariates, tau.regb and tau.regw for within- and between-study covariate effect, respectively. and tau.gamma for bias effect. The default of all heterogeneity parameters is ’dunif(0,2)’. Currently only the uniform distribution is supported. When the method.bias= ’adjust1’ or ’adjust2’, the user may provide priors to control the bias probability. For the bias probabilities, beta distributions are assumed with the following default values: RCT with low (pi.low.rct=’dbeta(1,10)’), high (pi.high.rct=’dbeta(10,1)’) bias, NRS with low (pi.low.rct=’dbeta(1,30)’) / high (pi.high.rct=’dbeta(30,1)’) bias (pi.low.nrs, pi.high.nrs).
`run.nrs`	An optional list is needed when the NRS used as a prior (method.bias=’prior’). The list consists of the following: (`var.infl`) controls the common inflation of the variance of NRS estimates (w) and its values range between 0 (NRS does not contribute at all and the prior is vague) and 1 (the NRS evidence is used at face value, default approach). The parameter (`mean.shift`) is the bias shift (ζ) to be added/subtracted from the estimated mean treatment effects (on the log-scale) from NRS network (0 is the default). `trt.effect` is a character indicates how to combine treatment effects across NRS studies .Options are ’random’ or ’common’ (default). Here you can also specify the arguments to control the MCMC chains with default value is in the parentheses: the number of adaptions n.adapt (500), number of iterations n.iter(10000), number of burn in n.burnin (4000), number of thinning thin (1) and number of chains n.chains (2). `jags.model` from rjags package describes these arguments.
`quiet`	A logical passed on to `jags.model`.

Details

This function creates a JAGS model and the needed data. The JAGS code is created from the internal function `crossnma.code`.

Covariates provided in arguments `cov1`, `cov2` and `cov3` can be either numeric or dichotomous (should be provided as factor or character) variables. By default, no covariate adjustment is applied (network meta-analysis).

Value

An object of class `crossnma.model` containing information on the JAGS model, which is a list containing the following components:

`model`	A long character string containing JAGS code that will be run in `jags.model`.

`data`	The data to be used to run JAGS model.
`trt.key`	A table of the treatments and its mapped integer number (as used in JAGS model).
`study.key`	A table of the studies and its mapped integer number (as used in JAGS model).
`trt.effect`	A character defining the model for the study-specific treatment effects.
`method.bias`	A character for defining the method to combine randomized clinical trials (RCT) and non-randomized studies (NRS).
`covariate`	A vector of the the names of the covariates (`cov1`, `cov2` and `cov3`) in prt.data and std.data used in network meta-regression.
`cov.ref`	A vector of values of `cov1.ref`, `cov2.ref`, `cov3.ref` to center continuous covariates. Dichotomous covariates take NA.
`dich.cov.labels`	A matrix with the levels of each dichotomous covariate and the corresponding assigned 0 / 1 values.
`split.regcoef`	A logical value. If FALSE the within- and between-study regression coefficients will be considered equal.
`regb.effect`	A character indicating the model for the between-study regression coefficients across studies.
`regw.effect`	A character indicating the model for the within-study regression coefficients across studies.
`bias.effect`	A character indicating the model for the bias coefficients across studies.
`bias.type`	A character indicating the effect of bias on the treatment effect; additive ('add') or multiplicative ('mult') or both ('both').
`all.data.ad`	A data.frame object with the prt.data (after it is aggregated) and std.data in a single dataset.
`call`	Function call.
`version`	Version of R package **crossnma** used to create object.

Author(s)

Tasnim Hamza `<tasnim.hamza@ispm.unibe.ch>`, Guido Schwarzer `<sc@imbi.uni-freiburg.de>`

See Also

`crossnma`, `jags.model`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data
```

```
# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")

# Fit JAGS model
# (suppress warning 'Adaptation incomplete' due to n.adapt = 20)
fit <-
  suppressWarnings(crossnma(mod, n.adapt = 20,
    n.iter = 50, thin = 1, n.chains = 3))

# Display the output
summary(fit)
plot(fit)
```

`heatplot.crossnma` *Heat Plot*

Description

Produces a heat plot that contain point estimates of relative effects for all possible pairs of treatments along with 95% credible intervals obtained with the quantile method.

Usage

```
## S3 method for class 'crossnma'
heatplot(
  x,
  median = TRUE,
  exp = TRUE,
  order = NULL,
  low.colour = "red",
  mid.colour = "white",
  high.colour = "springgreen4",
  cov1.value = NULL,
  cov2.value = NULL,
  cov3.value = NULL,
  size = 6,
  size.trt = 20,
  size.axis = 12,
  digits = 2,
  ...
)

heatplot(x, ...)
```

Arguments

`x`	An object created with `crossnma`.
`median`	A logical indicating whether to use the median (default) or mean to measure relative treatment effects.
`exp`	If TRUE (default), odds ratios are displayed. If FALSE, log odds ratios will be presented.
`order`	A vector of treatment names (character) representing the order in which to display these treatments.
`low.colour`	A string indicating the colour of low relative treatment effects for the heat plot (e.g odds ratio of ~0.5)
`mid.colour`	A string indicating the colour of null relative treatment effects for the heat plot (e.g odds ratio of ~1.0).
`high.colour`	A string indicating the colour of high relative treatment effects for the heat plot (e.g odds ratio of ~2.0).
`cov1.value`	The participant covariate value of `cov1` for which to report the results. Must be specified for network meta-regression and when individual participant dataset is used in the analysis. For dichotomous covariates, a character of the level (used in the data) should be indicated.
`cov2.value`	The participant covariate value of `cov2` for which to report the results. Must be specified for network meta-regression and when individual participant dataset is used in the analysis. For dichotomous covariates, a character of the level (used in the data) should be indicated.
`cov3.value`	The participant covariate value of `cov3` for which to report the results. Must be specified for network meta-regression and when individual participant dataset is used in the analysis. For dichotomous covariates, a character of the level (used in the data) should be indicated.
`size`	The size of cell entries with the relative treatment effect and 95% credible intervals.
`size.trt`	The size of treatment names placed on the top and left of the plot.
`size.axis`	The size of labels on the top and left of the plot
`digits`	The number of digits to be used when displaying the results.
`...`	Additional arguments (ignored at the moment).

Value

League heat plot, where a color scale is used to represent the relative treatment effects.

Author(s)

Tasnim Hamza `<tasnim.hamza@ispm.unibe.ch>`

See Also

`crossnma`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")

# Fit JAGS model
# (suppress warning 'Adaptation incomplete' due to n.adapt = 20)
fit <-
  suppressWarnings(crossnma(mod, n.adapt = 20,
    n.iter = 50, thin = 1, n.chains = 3))

# Create a heat plot
heatplot(fit)
```

`ipddata` *Simulated individual participant dataset.*

Description

A dataset containing 2950 participants who are treated in four different treatments: A, B, C and D. The dataset includes four studies. The outcome is binary. There are 10 attributes on individual level.

Usage

```
ipddata
```

Format

A data frame with 2950 rows and 10 variables:

id numeric, study identifier

relapse binary data, respond indicator, 0=no relapse and 1=relapse

treat character, indicating the assigned treatment to each participant

design character, design of the study, either ’rct’ or ’nrs’

age numeric, age of the participant

sex binary data, sex of the participant, 0=Female and 1=Male

rob character, the risk of bias of the study, ’low’, ’high’,’unclear’

unfavored numeric, the indicator of the unfavored treatment in each study, values are 0 or 1

bias.group numeric, the bias effect of the study, 1 = if the study has inactive treatment and adjust for bias effect, 2= if the study has active treatments and it is assumed another bias effect, 0=no bias adjustment

year numeric, the year study published

`league.crossnma` *League Table*

Description

Produces a league table that contains point estimates of relative effects for all possible pairs of treatments along with 95% credible intervals obtained with the quantile method.

Usage

```
## S3 method for class 'crossnma'
league(
  x,
  median = TRUE,
  exp = FALSE,
  order = NULL,
  cov1.value = NULL,
  cov2.value = NULL,
  cov3.value = NULL,
  digits = 2,
  direction = "wide",
  ...
)

league(x, ...)

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

Arguments

`x`	An object created with `crossnma`.
`median`	A logical indicating whether to use the median (default) or mean to measure relative treatment effects.
`exp`	If TRUE (default), odds ratios are displayed. If FALSE, log odds ratios will be presented.
`order`	A vector of treatment names (character) representing the order in which to display these treatments.

`cov1.value`	The participant covariate value of `cov1` for which to report the results. Must be specified for network meta-regression and when individual participant dataset is used in the analysis. For dichotomous covariates, a character of the level (used in the data) should be indicated.
`cov2.value`	The participant covariate value of `cov2` for which to report the results. Must be specified for network meta-regression and when individual participant dataset is used in the analysis. For dichotomous covariates, a character of the level (used in the data) should be indicated.
`cov3.value`	The participant covariate value of `cov3` for which to report the results. Must be specified for network meta-regression and when individual participant dataset is used in the analysis. For dichotomous covariates, a character of the level (used in the data) should be indicated.
`digits`	The number of digits to be used when displaying the results.
`direction`	The format to display the league table. Two options "wide" (default) and "long".
`...`	Additional arguments (ignored at the moment).

Value

A league table. Row names indicate comparator treatments. The table will be displayed in a long or wide formatting.

Author(s)

Tasnim Hamza <`tasnim.hamza@ispm.unibe.ch`>

See Also

`crossnma`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = ”A”, trt.effect = ”random”, method.bias = ”naive”)

# Fit JAGS model
# (suppress warning 'Adaptation incomplete' due to n.adapt = 20)
fit <-
  suppressWarnings(crossnma(mod, n.adapt = 20,
    n.iter = 50, thin = 1, n.chains = 3))
```

```
# Create league tables
league(fit, exp = TRUE)                        #  wide format
league(fit, exp = TRUE, direction = "long") #  long format
```

`netgraph.crossnma` *Produce a network plot*

Description

Create a network plot of the cross network meta-analysis or meta-regression

Usage

```
## S3 method for class 'crossnma'
netgraph(x, ...)
```

Arguments

`x`	An object produced by `crossnma`.
`...`	... Additional arguments (passed on to `netgraph.netmeta`)

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)

Tasnim Hamza <`tasnim.hamza@ispm.unibe.ch`>

See Also

`netgraph.netmeta`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")

# Fit JAGS model
# (suppress warning 'Adaptation incomplete' due to n.adapt = 20)
fit <-
  suppressWarnings(crossnma(mod, n.adapt = 20,
    n.iter = 50, thin = 1, n.chains = 3))

# Create network plot
netgraph(fit)
```

`netgraph.crossnma.model`

Produce a network plot

Description

Create a network plot of the cross network meta-analysis or meta-regression

Usage

```
## S3 method for class 'crossnma.model'
netgraph(x, ...)
```

Arguments

`x`	An object produced by `crossnma.model`.
`...`	... Additional arguments (passed on to `netgraph.netmeta`)

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)

Tasnim Hamza <`tasnim.hamza@ispm.unibe.ch`>

See Also

`netgraph.netmeta`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")

# Fit JAGS model
# (suppress warning 'Adaptation incomplete' due to n.adapt = 20)
fit <-
  suppressWarnings(crossnma(mod, n.adapt = 20,
    n.iter = 50, thin = 1, n.chains = 3))

# Create network plot
netgraph(mod)
```

`plot.crossnma` *Trace plot of MCMC output*

Description

Produces a separate plot for each parameter in the JAGS model. Each plot shows iterations vs sampled values.

Usage

```
## S3 method for class 'crossnma'
plot(x, ...)
```

Arguments

`x`	An object generated by `crossnma`.
`...`	Additional arguments (passed on to `traceplot`)

Value

No return value (plot function).

Author(s)

Tasnim Hamza <tasnim.hamza@ispm.unibe.ch>

See Also

`crossnma`, `traceplot`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")

# Fit JAGS model
# (suppress warning 'Adaptation incomplete' due to n.adapt = 20)
fit <-
  suppressWarnings(crossnma(mod, n.adapt = 20,
    n.iter = 50, thin = 1, n.chains = 3))

# Trace plot of MCMC output
plot(fit)
```

`print.crossnma` *Print call used to create JAGS model for cross-design & -format network meta-analysis or regression*

Description

Print call used to create JAGS model for cross-design & -format network meta-analysis or regression

Usage

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

Arguments

`x`	An object of class `crossnma`.
`...`	Additional arguments (ignored).

Value

No return value (print function).

Author(s)

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

See Also

`crossnma`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")

# Fit JAGS model
# (suppress warning 'Adaptation incomplete' due to n.adapt = 20)
fit <-
  suppressWarnings(crossnma(mod, n.adapt = 20,
    n.iter = 50, thin = 1, n.chains = 3))
```

```
fit
```

---

print.crossnma.model *Print call used to create JAGS model for cross-design & -format network meta-analysis or regression*

Description

Print call used to create JAGS model for cross-design & -format network meta-analysis or regression

Usage

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

Arguments

`x`	An object of class `crossnma.model`.
`...`	Additional arguments (ignored).

Value

No return value (print function).

Author(s)

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

See Also

`crossnma.model`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")
mod
```

`print.summary.crossnma`

Print summary of cross-design & -format network meta-analysis or regression

---

Description

Print summary results for cross-design and cross-format network meta-analysis or meta-regression

Usage

```
## S3 method for class 'summary.crossnma'
print(x, digits = 3, ...)
```

Arguments

`x`	An object of class `crossnma`.
`digits`	The number of significant digits printed. The default value is 3.
`...`	Additional arguments.

Value

No return value (print function).

Author(s)

Tasnim Hamza <`tasnim.hamza@ispm.unibe.ch`>

See Also

`summary.crossnma`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")

# Fit JAGS model
# (suppress warning 'Adaptation incomplete' due to n.adapt = 20)
```

```
fit <-
  suppressWarnings(crossnma(mod, n.adapt = 20,
    n.iter = 50, thin = 1, n.chains = 3))

# Display the output (with 5 digits)
print(summary(fit), digits = 5)
```

`print.summary.crossnma.model` *Print code of JAGS model for cross-design & -format network meta-analysis or regression*

Description

Print code of JAGS model for cross-design & -format network meta-analysis or regression

Usage

```
## S3 method for class 'summary.crossnma.model'
print(x, ...)
```

Arguments

`x`	An object of class `summary.crossnma.model`.
`...`	Additional arguments (ignored).

Value

No return value (print function).

Author(s)

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

See Also

`summary.crossnma.model`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
```

```
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")
summary(mod)
```

`stddata` *Simulated aggregate dataset.*

Description

The dataset includes two randomized-controlled trials (RCTs), comparing treatments A and C. The outcome is binary represented as the number of participants with at least one relapse.

Usage

```
stddata
```

Format

A data frame with 4 rows and 11 variables:

id numeric, study identifier

relapse numeric, the number of relapses

n numeric, the sample size

treat character, indicating the assigned treatment to participants in each study arm

design character, design of the study, either 'rct' or 'nrs'

age numeric, the mean age of participants in each study

sex numeric, the proportion of females on each study

rob character, the risk of bias of the study, 'low', 'high','unclear'

unfavored numeric, the indicator of the unfavored treatment in each study, values are 0 or 1

bias.group numeric, the bias effect of the study, 1 = study has inactive treatment and adjust for bias effect, 2= study has active treatments and another adjustment for bias effect, 0=no bias adjustment

year numeric, the year published of the study

`summary.crossnma` *Summary function for crossnma object*

Description

This function creates posterior summary statistics for the fitted cross network meta-analysis / meta-regression model

Usage

```
## S3 method for class 'crossnma'
summary(object, quantiles = c(0.025, 0.5, 0.975), exp = TRUE, ...)
```

Arguments

`object`	An object generated by the `crossnma`.
`quantiles`	A numeric vector of probabilities to present posterior summaries. The default value is c(0.025, 0.5, 0.975) for the 95% credible interval and the median.
`exp`	A logical value indicating whether to exponentiate the parameters of relative treatment effect and covariate effect. Default is TRUE.
`...`	Additional arguments to be passed to summary() function

Value

`crossnma.summary` returns a matrix containing the following summary statistics (in columns) for each estimated parameter:

`Mean` the mean of the posterior distribution

`SD` the standard deviation of the posterior distribution

`2.5%` (default) the 2.5% quantile of the posterior distribution (the lower bound of the 95% credible interval)

`50%` (default) the median of the posterior distribution

`97.5%` (default) the 97.5% quantile of the posterior distribution (the upper bound of the 95% credible interval)

`Rhat` Gelman-Rubin statistic. The further the value of Rhat from 1, the worse the mixing of chains and so the convergence.

`n.eff` An estimate of the effective sample size. The smaller the value of n.eff the greater the uncertainty associated with the corresponding parameter.

Author(s)

Tasnim Hamza <`tasnim.hamza@ispm.unibe.ch`>

See Also

`print.summary.crossnma`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random", method.bias = "naive")

# Fit JAGS model
# (suppress warning 'Adaptation incomplete' due to n.adapt = 20)
fit <-
  suppressWarnings(crossnma(mod, n.adapt = 20,
    n.iter = 50, thin = 1, n.chains = 3))

# Display the output
summary(fit)
```

`summary.crossnma.model` *Summary function for crossnma.model object*

Description

Summary function for crossnma.model object

Usage

```
## S3 method for class 'crossnma.model'
summary(object, ...)
```

Arguments

`object`	An object generated by the `crossnma.model`.
`...`	Additional arguments (ignored)

Value

An object of classes `summary.crossnma.model` and `crossnma.model` containing information on the JAGS model, which is a list containing the following components:

`model`	A long character string containing JAGS code that will be run in `jags.model`.
`data`	The data to be used to run JAGS model.

`trt.key`	A table of the treatments and its mapped integer number (as used in JAGS model).
`study.key`	A table of the studies and its mapped integer number (as used in JAGS model).
`trt.effect`	A character defining the model for the study-specific treatment effects.
`method.bias`	A character for defining the method to combine randomized clinical trials (RCT) and non-randomized studies (NRS).
`covariate`	A vector of the the names of the covariates (`cov1`, `cov2` and `cov3`) in prt.data and std.data used in network meta-regression.
`cov.ref`	A vector of values of `cov1.ref`, `cov2.ref`, `cov3.ref` to center continuous covariates. Dichotomous covariates take NA.
`dich.cov.labels`	A matrix with the levels of each dichotomous covariate and the corresponding assigned 0 / 1 values.
`split.regcoef`	A logical value. If FALSE the within- and between-study regression coefficients will be considered equal.
`regb.effect`	A character indicating the model for the between-study regression coefficients across studies.
`regw.effect`	A character indicating the model for the within-study regression coefficients across studies.
`bias.effect`	A character indicating the model for the bias coefficients across studies.
`bias.type`	A character indicating the effect of bias on the treatment effect; additive ('add') or multiplicative ('mult') or both ('both').
`all.data.ad`	A data.frame object with the prt.data (after it is aggregated) and std.data in a single dataset.
`call`	Function call.
`version`	Version of R package **crossnma** used to create object.

Author(s)

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

See Also

`print.summary.crossnma.model`

Examples

```
# We conduct a network meta-analysis assuming a random-effects
# model.
# The data comes from randomized-controlled trials and
# non-randomized studies (combined naively)
head(ipddata) # participant-level data
head(stddata) # study-level data

# Create a JAGS model
mod <- crossnma.model(treat, id, relapse, n, design,
```

```
    prt.data = ipddata, std.data = stddata,
    reference = "A", trt.effect = "random", method.bias = "naive")
summary(mod)
```

Index