Package

August 31, 2021

Title Bayesian network meta-analyses in compliance with best practice and reporting guidelines

Version 1.1.0

Description BUGSnet (Bayesian inference Using Gibbs Sampling to conduct NETwork meta-analysis) is a feature-rich R package to conduct Bayesian network meta-analyses in compliance with best practice and reporting guidelines. Bayesian analyses are conducted with JAGS. Outputs are highly customizable and include network plots, tables of network characteristics, league tables and league heat plots, SUCRA plots, rankograms, forest plots, leverage plots, traceplots, posterior mean deviance comparison plots.

Depends R (>= 3.6),
rjags (>= 4.9)

License CC BY-NC-SA 4.0

Encoding UTF-8

LazyData yes

RoxygenNote 7.1.1

Imports tidyr (>= 1.0),
plyr (>= 1.8.4),
ggplot2 (>= 3.2.1),
igraph (>= 1.2.4.1),
extrafont (>= 0.17),
magrittr (>= 1.5),
dplyr (>= 0.8.3),
meta (>= 4.9.7),
purrr (>= 0.3.3),
scales (>= 1.0.0),
RColorBrewer (>= 1.1.2),
Rdpack (>= 0.11.0),
graphics (>= 3.6.1),
utils (>= 3.6.1),
rlang (>= 0.4.1),
tibble (>= 2.1.3),
stringr (>= 1.4.0),
gridExtra (>= 2.3),
mcmcr

Suggests knitr (>= 1.25),
rmarkdown (>= 1.16),

devtools (>= 2.2.1),
roxygen2 (>= 6.1.1),
testthat (>= 2.2.1)

VignetteBuilder knitr

Roxygen list(markdown = TRUE, roclets = c(``rd'', ``namespace'', ``collate''))

RdMacros Rdpack

R topics documented:

`afib` *Atrial Fibrillation data*

Description

The atrial fibrillation data was previously used to demonstrate the network meta-analysis capabilities of the R package 'gemtc'. This dataset was originally generated by Cooper et al. (2009) and was subsequently revised by van Valkenhoef et al (2013). This dataset provides information from 25 clinical trials examining the effects of various interventions (17) on the risk of stroke (binary outcome) among individuals with non-rheumatic atrial fibrillation. The stroke variable is a covariate indicating the proportion of patients in the study that had a prior stroke.

Usage

```
data(afib)
```

Format

A dataset.

References

Cooper NJ, Sutton AJ, Morris D, Ades AE, Welton NJ (2009). "Adressing between-study heterogeneity and inconsistency in mixed treatment comparisons: Application to stroke prevention treatments in individuals with non-rheumatic atrial fibrillation." *Statistics in Medicine*, **28**, 1861-1881.

van Valkenhoef, G, Lu, G, de Brock, B, Hillege, H, Ades, AE, Welton, NJ (2012). "Automating network meta-analysis." *Research Synthesis Methods*, **3**(4), 285-299.

Examples

```
data(afib)
```

`BUGSnet` *BUGSnet: Bayesian network meta-analysis in compliance with best practice guidelines.*

Description

BUGSnet (Bayesian inference Using Gibbs Sampling to conduct NETwork meta-analysis) is a feature-rich R package to conduct Bayesian network meta-analyses in compliance with best practice and reporting guidelines. Bayesian analyses are conducted with JAGS and BUGS code is automatically generated by the package based on the user's inputs. Outputs are highly customizable and include network plots, tables of network characteristics, league tables and league heat plots, SUCRA plots, rankograms, forest plots, leverage plots, traceplots, posterior mean deviance comparison plots.

Details

To be able to use this package, you must have installed JAGS (Just Another Gibbs Sampler) on your computer.

How to cite this work? Type citation("BUGSnet").

References

Plummer M (2017). *JAGS Version 4.3.0 user manual*. http://people.stat.sc.edu/hansont/stat740/jags_user_manual.pdf, last accessed 2019-03-06.

Plummer M (2017). "JAGS download page." http://mcmc-jags.sourceforge.net/, last accessed 2019-03-06.

`data.plot` *Patient Characteristic Plot*

Description

Plots a particular patient characteristic by study or by treatment. Useful for assessing differences in potential effect modifiers.

Usage

```
data.plot(
  data,
  covariate,
  half.length = NULL,
  by = "study",
  avg.hline = TRUE,
  fill.str = NULL,
  text.size = 20
)
```

Arguments

`data`	A `BUGSnetData` object produced by `data.prep()`
`covariate`	A string indicating the name of the patient characteristic to be plotted
`half.length`	A string indicating how to calculate the half-length of error bars (optional)
`by`	If by="study" then data from arms will be grouped by study/trial. If by="treatment" then bar graph is grouped by treatment.
`avg.hline`	If TRUE, adds overall average line to plot. Default is TRUE.
`fill.str`	An optional string indicating the variable to categorize measurements. For instance, some studies report the mean treatment effect and others may report the median treatment effect. If there is a variable in `data` called "type.measure" indicating whether the mean or median is reported, setting fill.str="type.measure" would colour all studies reporting the mean as red, and all the studies reporting the median as turquoise.
`text.size`	Font size of the text. Default is 20.

See Also

`data.prep`

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

# Example containing a fill.str, an overall average, and no error
```

```
data.plot(data = diabetes.slr,
             covariate = "age",
             fill.str="age_type",
             by = "study")

# Example containing no fill.str, no overall average, but contains errorbars

data.plot(data = diabetes.slr,
             covariate = "age",
             half.length = "age_SD",
             avg.hline=FALSE,
             by = "study")
```

`data.prep` *Data preparation*

Description

Puts data into appropriate form for subsequent analysis using BUGSnet.

Usage

```
data.prep(arm.data, varname.t, varname.s)
```

Arguments

`arm.data`	Data with 1 row for each study arm.
`varname.t`	A string indicating the name of the treatment variable.
`varname.s`	A string indicating the name of the study variable.

Value

`data.prep` returns an object of class `BUGSnetData` which is a list containing the following components:

`arm.data` - A tibble containing the arm level study data

`treatments` - A list of all treatments in the network

`studies` - A list of all studies in the network

`n.arms` - A tibble containing the number of arms for each study

`varname.t` - A string containing the name of the treatment variable

`varname.s` - A string containing the name of the study variable

See Also

`net.plot`, `net.tab`, `data.plot`, `nma.model`

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

diabetes.slr$arm.data
diabetes.slr$treatments
diabetes.slr$studies
```

---

diabetes *Diabetes data*

Description

The diabetes dataset has been used to illustrate network meta-analysis within the NICE-DSU TSD 2 (examples 3a and 3b). The dataset is comprised of information from 22 clinical trials examining the effect of antihypertensive drugs (there were six treatments including a placebo arm) on the risk of developing diabetes (a binary outcome). This dataset also contains information on the years of follow-up within each arm.

Usage

```
data(diabetes)
```

Format

A dataset.

References

Dias S, Welton NJ, Sutton AJ, Ades AE (2011). "NICE DSU technical support document 2: a generalised linear modelling framework for pairwise and network meta-analysis of randomised controlled trials." National Institute for Health and Clinical Excellence.

Elliott WJ, Meyer PM (2007). "Incident diabetes in clinical trials of antihypertensive drugs: a network meta-analysis." *The Lancet*, **369**(9557), 201–207.

Examples

```
data(diabetes)
```

diabetes.sim *Diabetes data with simulated age covariate*

Description

This is the same as the Diabetes dataset but we have added a simulated age variable to this dataset for the purpose of demonstrating certain features within BUGSNET.

Usage

```
data(diabetes.sim)
```

Format

A dataset.

References

Dias S, Welton NJ, Sutton AJ, Ades AE (2011). "NICE DSU technical support document 2: a generalised linear modelling framework for pairwise and network meta-analysis of randomised controlled trials." National Institute for Health and Clinical Excellence.

Elliott WJ, Meyer PM (2007). "Incident diabetes in clinical trials of antihypertensive drugs: a network meta-analysis." *The Lancet*, **369**(9557), 201–207.

Examples

```
data(diabetes.sim)
```

net.plot *Network Plot*

Description

Produces network plot where nodes represent treatments and edges represent direct evidence (e.g an RCT) comparing treatments.

Usage

```
net.plot(
  data,
  node.scale = 5,
  edge.scale = 2,
  flag = NULL,
  study.counts = FALSE,
  label.offset1 = 0,
  label.offset2 = 1,
  graph.scale = TRUE,
  node.lab.cex = 1,
  edge.lab.cex = 1,
```

```
  node.colour = "#f69c54",
  edge.colour = "grey",
  edge.lab.colour = "blue",
  flag.edge.colour = "lightpink",
  layout = "layout_in_circle",
  layout.params = NULL
)
```

Arguments

data	A BUGSnetData object produced by data.prep().
node.scale	Size of the nodes (default=5)
edge.scale	Thickness of the edges (default=2).
flag	Used to highlight direct comparisons to particular treatments (optional). Set this value to treatment(s) of interest and it will highlight, in red, all of the edges going into the specified treatment(s).
study.counts	If TRUE, prints the number of studies on each edge.
label.offset1	Node label location (x-axis) relative to node. Default=0
label.offset2	Node label location (y-axis) relative to node. Default=1
graph.scale	Whether to make edges and nodes proportionnaly larger with the number of studies/arms. Default is TRUE.
node.lab.cex	Size of node labels
edge.lab.cex	Size of edge labels
node.colour	Node colour (string)
edge.colour	Edge colour (string)
edge.lab.colour	Edge label colour (string)
flag.edge.colour	Color of flagged edges (string)
layout	Specifies how the nodes should be layed out in the graph. Default is "layout_in_circle". See igraph::layout_ for layout options
layout.params	Additional parameters to be passed to the chosen 'layout' function. See igraph::layout_ for more information

See Also

data.prep, igraph::layout_

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

# use default settings
net.plot(diabetes.slr)
```

```
# Highlight all direct comparisons with Placebo. Adjust node and edge size, centre node labels
net.plot(diabetes.slr,
node.scale=4,
edge.scale=1.5,
flag="Placebo",
label.offset1=0,
label.offset2=0)
```

---

net.tab *Generate Network Characteristics*

Description

Generates tables of network characteristics

Usage

```
net.tab(data, outcome, N, type.outcome, time = NULL)
```

Arguments

data	A BUGSnetData object produced by data.prep()
outcome	A string indicating the name of the outcome variable
N	A string indicating the name of the variable containing the number of participants in each arm
type.outcome	A string. Options are: "binomial", "continuous", "rate" (e.g # of events and # person-time reported), "rate2" (e.g # events and followup time reported)
time	A string required when type.outcome = "rate" or "rate2". Name of variable indicating person-time followup (e.g person years) or study followup time

Value

network - Table of network characteristics

intervention - Summary statistics broken down by treatment

comparison - Summary statistics broken down by treatment comparison

See Also

data.prep

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

network.char <- net.tab(data = diabetes.slr,
outcome = "diabetes",
N = "n",
type.outcome = "rate2",
time = "followup")
```

nma.compare *Consistency vs Inconsistency plot*

Description

Plots the posterior mean deviance of a consistency model vs an inconsistency model. This plot can help identify loops where inconsistency is present. Ideally, both models will contribute approximately 1 to the posterior mean deviance.

Usage

```
nma.compare(consistency.model.fit, inconsistency.model.fit, ...)
```

Arguments

consistency.model.fit	Results of nma.fit() of an consistency model.
inconsistency.model.fit	Results of nma.fit() of an inconsistency model.
...	Graphical arguments such as main=, ylab=, and xlab= may be passed in plot().

See Also

nma.run

Examples

```
data(diabetes.sim)
diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

#Random effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.re.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "random",
       type="consistency",
       time="followup"
       )

diabetes.re.c.res <- nma.run(diabetes.re.c,
n.adapt=100,
n.burnin=0,
n.iter=100)

#Random effects, inconsistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.re.i <- nma.model(data = diabetes.slr,
```

```
        outcome = "diabetes",
        N = "n",
        reference = "Placebo",
        family = "binomial",
        link = "cloglog",
        effects = "random",
        type="inconsistency",
        time="followup"
        )

diabetes.re.i.res <- nma.run(diabetes.re.i,
n.adapt=100,
n.burnin=0,
n.iter=100)

# Assess model fit for a both an inconsistency model and consistency model using nma.fit()
assess.consistency <- nma.fit(diabetes.re.c.res)
assess.inconsistency <- nma.fit(diabetes.re.i.res)

#Plot the results against each other to assess inconsistency
nma.compare(assess.consistency, assess.inconsistency)
```

`nma.diag` *Trace plots and convergence diagnostics for MCMC chains*

Description

Produces trace plots and Gelman-Rubin and Geweke convergence diagnostics for the MCMC chains obtained from `nma.run()`. The Gelman-Rubin and Geweke diagnostics are implemented using functions from the coda package.

Usage

```
nma.diag(
  nma,
  trace = TRUE,
  gelman.rubin = TRUE,
  geweke = TRUE,
  params = "all",
  thin = 1,
  ncol = 1,
  nrow = 3,
  plot_prompt = TRUE,
  geweke_frac1 = 0.1,
  geweke_frac2 = 0.5
)
```

Arguments

`nma`	A `BUGSnetRun` object produced by `nma.run()`
`trace`	If TRUE, outputs trace plots. Default is TRUE.
`gelman.rubin`	If TRUE, runs Gelman-Rubin diagnostic. Default is TRUE.

geweke	If TRUE, runs Geweke diagnostic. Default is TRUE.
params	Integer or character vector which specifies which parameters to produce trace plots for when trace is set to TRUE. Default is "all" which plots every monitored parameter.
thin	Thinning factor for the mcmc chains when producing trace plots. Default is 1.
ncol	Number of columns in each batch of trace plots
nrow	Number rows in each batch of trace plots
plot_prompt	If TRUE, prompts the user to hit enter before plotting each additional batch of trace plots. Default is TRUE.
geweke_frac1	Fraction to use from beginning of chain. Default is 0.1.
geweke_frac2	Fraction to use from end of chain. Default is 0.5.

Value

`gelman.rubin` An object of class `gelman.rubin.results` containing the Gelman-Rubin diagnostic results. A formatted table with custom PSRF threshold can be printed using `print(x,gelman.rubin.threshold = 1.2)`.

`geweke` An object of class `geweke.results` containing the Geweke diagnostic results. A formatted table with custom significance level can be printed using `print(x,alpha = 0.05)`.

See Also

`nma.run`

Examples

```
data(thrombolytic)
dich.slr <- data.prep(arm.data = thrombolytic, varname.t = "treatment",
                      varname.s = "study")
random_effects_model <- nma.model(data=dich.slr, outcome="events",
                                  N="sampleSize", reference="SK",
                                  family="binomial", link="log",
                                  effects="random")
random_effects_results <- nma.run(random_effects_model, n.adapt=100,
                                  n.burnin=0, n.iter=100)
nma.diag(random_effects_results)
```

---

nma.fit *Assess Model Fit*

Description

Computes the Deviance Information Criteria (DIC) and produces a leverage plot (as in the NICE Technical Support Document 2) for a given model. These can be used to assess and compare the fit of different models (i.e fixed vs random effects, consistency vs inconsistency). `nma.fit` also produces a plot comparing the leverage of each data point against their contribution to the total posterior deviance. Points lying outside the purple dotted line are generally identified as contributing to the model's poor fit. Points with high leverage are influencial i.e. they have a stong influence on the estimates.

Usage

```
nma.fit(nma, plot.pD = TRUE, plot.DIC = TRUE, plot.Dres = TRUE, ...)
```

Arguments

nma	A BUGSnetRun object produced by running nma.run().
plot.pD	Whether to include pD on the plot. Default is TRUE.
plot.DIC	Whether to include DIC on the plot. Default is TRUE.
plot.Dres	Whether to include Dres on the plot. Default is TRUE.
...	Graphical arguments such as main=, ylab=, and xlab= may be passed as in plot(). These arguments will only effect the leverage plot.

Value

DIC - A number indicating the Deviance Information Criteria. The DIC is calculated as the sum of Dres and pD. A larger DIC is indicative of a worse model fit.

leverage - A vector with one value per study arm indicating the leverage of each data point (study arm). Leverage is defined as pmdev minus the deviance at the posterior mean of the fitted values.

w - A vector with one value per study arm. The magnitude of w represents the data point's contribution to the posterior mean deviance of the model and is simply the square root of pmdev. The sign indicates whether the data point is being over (negative sign) or under (positive sign) estimated by the model and is calculated as the sign of the difference of the observed outcome minus the predicted outcome.

pmdev - A vector with one value per study arm representing the posterior mean residual deviance for each data point (study arm).

Dres - The posterior mean of the residual deviance.

pD - The effective number of parameters, calculated as the sum of the leverages.

See Also

nma.run

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

#Random effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.re.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "random",
       type="consistency",
       time="followup"
```

```
       )

diabetes.re.c.res <- nma.run(diabetes.re.c,
n.adapt=100,
n.burnin=0,
n.iter=100)

#Fixed effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.fe.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "fixed",
       type="consistency",
       time="followup"
       )

diabetes.fe.c.res <- nma.run(diabetes.fe.c,
n.adapt=100,
n.burnin=0,
n.iter=100)

#Compare fixed vs random effects via leverage plots and DIC
par(mfrow=c(1,2))
nma.fit(diabetes.fe.c.res, main = "Fixed Effects Model" )
nma.fit(diabetes.re.c.res, main= "Random Effects Model")
```

`nma.forest` *Forest Plot*

Description

Produces a forest plot of point estimates and 95% credible intervals obtained with the quantile method.

Usage

```
nma.forest(
  nma,
  comparator,
  central.tdcy = "median",
  order = NULL,
  log.scale = FALSE,
  lwd = 1,
  x.trans = NULL,
  cov.value = NULL
)
```

Arguments

nma	A BUGSnetRun object produced by running nma.run().
comparator	The treatment to use as a comparator
central.tdcy	The posterior statistic used in order to measure relative effectiveness. The options are "mean" and "median". Default is median.
order	Optional. A vector of strings representing the order in which to display the treatments.
log.scale	If TRUE, odds ratios, relative risk or hazard ratios are reported on the log scale. Default is FALSE.
lwd	Line width relative to the default (default=1).
x.trans	Optional. A string indicating a transformation to apply to the x-axis. Setting this parameter to "log" is useful when there are extreme values or to allow an easier interpretation of odds ratios and relative ratios (if e.g. treatment B is twice as far from the line y=1 then treatment A then it's OR/RR is twice that of treatment A.)
cov.value	Must be specified for meta-regression. This is the value of the covariate for which to report the results.

Value

forestplot - A forest plot.

See Also

nma.run, nma.league, nma.rank

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

#Random effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.re.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "random",
       type="consistency",
       time="followup"
       )

diabetes.re.c.res <- nma.run(diabetes.re.c,
n.adapt=100,
n.burnin=0,
n.iter=100)
```

```
#make forest plot
nma.forest(nma = diabetes.re.c.res, comparator="Placebo")
```

`nma.league` *League Table and Heat Plot*

Description

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

Usage

```
nma.league(
  nma,
  central.tdcy = "median",
  log.scale = FALSE,
  order = NULL,
  low.colour = "darkgoldenrod1",
  mid.colour = "white",
  high.colour = "cornflowerblue",
  cov.value = NULL,
  digits = 2
)
```

Arguments

`nma`	A `BUGSnetRun` object produced by running `nma.run()`.
`central.tdcy`	The statistic that you want to use in order to measure relative effectiveness. The options are "mean" and "median".
`log.scale`	If TRUE, odds ratios, relative risk or hazard ratios are reported on the log scale. Default is FALSE.
`order`	A vector of strings representing the order in which to display the treatments.
`low.colour`	A string indicating the colour of low relative treatment effects for the heat plot (e.g relative risk of 0.5).
`mid.colour`	A string indicating the colour of null relative treatment effects for the heat plot (e.g relative risk of ~1.0).
`high.colour`	A string indicating the colour of high relative treatment effects for the heat plot (e.g relative risk of ~2.0).
`cov.value`	Must be specified for meta-regression. This is the value of the covariate for which to report the results.
`digits`	Number of digits to display after the decimal point

Value

`table` - A league table. Row names indicate comparator treatments.

`longtable` - League table in the long format.

`heatplot` - League heat plot, where a color scale is used to represent relative treatment effects and ** are used to highlight statistically significant differences.

See Also

nma.run, nma.rank, nma.forest

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

#Random effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.re.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "random",
       type="consistency",
       time="followup"
       )

diabetes.re.c.res <- nma.run(diabetes.re.c,
n.adapt=100,
n.burnin=0,
n.iter=100)


league_table <- nma.league(nma=diabetes.re.c.res, central.tdcy="median")
league_table$heatplot
league_table$table
league_table$longtable
```

---

nma.model *Create Bugs Model for Arm-Level Data*

Description

Creates BUGS code which can be ran through nma.run().

Usage

```
nma.model(
  data = NULL,
  outcome,
  N = NULL,
  sd = NULL,
  reference,
  type = "consistency",
  time = NULL,
  family = NULL,
```

```
  link = NULL,
  effects,
  prior.mu = "DEFAULT",
  prior.d = "DEFAULT",
  prior.sigma = "DEFAULT",
  prior.beta = NULL,
  covariate = NULL
)
```

Arguments

data	A BUGSnetData object containing the data from arm-based trials produced by data.prep()
outcome	A string indicating the name of your outcome variable.
N	A string indicating the name of the variable containing the number of participants in each arm.
sd	A string (only required for continuous outcomes) indicating variable name of the standard deviation of the outcome. Standard errors should be converted to standard deviation by multiplying by the square root of the sample size prior to using this function.
reference	A string for the treatment that will be seen as the 'referent' comparator and labeled as treatment 1 in the BUGS code. This is often a placebo or control drug of some kind.
type	If type="inconsistency", an inconsistency model will be built. By default, type="consistency" and a consistency model is built. will be built.
time	A string (only required for binomial-cloglog or poisson-log models) indicating the name of variable indicating person-time followup (e.g person years) or study followup time.
family	A string indicating the family of the distribution of the outcome. Options are: "binomial", "normal", "poisson".
link	The link function for the nma model. Options are "logit" (binomial family), "log" (binomial or poisson family), "cloglog" (binomial family), "identity" (normal family).
effects	A string indicating the type of treatment effect relative to baseline. Options are "fixed" or "random".
prior.mu	A string of BUGS code that defines priors on the baseline treatment effects. By default, independent normal priors are used with mean 0 and standard deviation 15u, where u is the largest maximum likelihood estimator in single trials (see van Valkenhoef, G et al. 2012).
prior.d	A string of BUGS code that defines define priors on relative treatment effects. By default, independent normal priors are used with mean 0 and standard deviation 15u, where u is the largest maximum likelihood estimator in single trials (see van Valkenhoef, G et al. 2012).
prior.sigma	A string of BUGS code that defines the prior on the variance of relative treatment effects. By default, a uniform distribution with range 0 to u is used, where u is the largest maximum likelihood estimator in single trials (see van Valkenhoef, G et al. 2012).
prior.beta	Optional string that defines the prior on the meta-regression coefficients. Options are "UNRELATED", "EXCHANGEABLE", "EQUAL" (TSD3) or a string of BUGS code.

`covariate` Optional string indicating the name of the variable in your data set that you would like to adjust for via meta regression. By default, covariate=NULL and no covariate adjustment is applied. If the specified covariate is numeric then it will be centered for the analysis. If it is a character or factor then it will be treated as categorical. Currently only categorical variables with fewer than 3 levels are supported.

Details

For meta-regression, the prespecified prior choices for the regression coefficients $\beta_{(1,2)}, \ldots, \beta_{(1,K)}$ are

Unrelated:

$$iidt(0, u^2, 1)$$

Exchangeable:

$$iidN(b, \gamma^2), b\, t(0, u^2, 1), \gamma\, U(0, u)$$

Equal:

$$\beta_2 = ... = \beta_T = B, B\, t(0, u^2, 1)$$

where u is the largest maximum likelihood estimator in single trials (see van Valkenhoef, G et al. 2012).

Value

`nma.model` returns an object of class `BUGSnetModel` which is a list containing the following components:

`bugs` - A long character string containing BUGS code that will be run in `jags`.

`data` - The data used in the BUGS code.

`scale` - The scale of the outcome, based on the chosen family and link function examples are "Risk Ratio" (relative risk), "Odds Ratio", "Mean Difference", "Hazard Ratio"

`trt.key` - Treatments mapped to integer numbers, used to run BUGS code.

...

References

van Valkenhoef, G, Lu, G, de Brock, B, Hillege, H, Ades, AE, Welton, NJ (2012). "Automating network meta-analysis." *Research Synthesis Methods*, **3**(4), 285-299.

Dias S, Sutton AJ, Welton NJ, Ades AE (2011). "NICE DSU technical support document 3. Heterogeneity: subgroups, meta-regression, bias and bias-adjustment." National Institute for Health and Clinical Excellence. Last updated April 2012.

See Also

`data.prep`, `nma.run`

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

#Random effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.re.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "random",
       type="consistency",
       time="followup"
       )

#Fixed effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.fe.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "fixed",
       type="consistency",
       time="followup"
       )
```

`nma.model.contrast` *Create Bugs Model for contrast-Level data*

Description

Creates BUGS code which can be ran through `nma.run()`.

Usage

```
nma.model.contrast(
  data_contrast = NULL,
  differences,
  se.diffs,
  var.arm1 = NULL,
  reference,
  type = "consistency",
  effects,
  scale,
```

```
  prior.mu = "DEFAULT",
  prior.d = "DEFAULT",
  prior.sigma = "DEFAULT"
)
```

Arguments

data_contrast	A BUGSnetData object containing the data from contrast-based trials produced by data.prep()
differences	A string indicating the name of the differences for contrast-based studies
se.diffs	A string indicating the variable name of the standard errors of the differences.
var.arm1	A string (only required for networks with multi-arm trials) indicating the variable name of the variance of the treatment in arm 1 of each study
reference	A string for the treatment that will be seen as the 'referent' comparator and labeled as treatment 1 in the BUGS code. This is often a placebo or control drug of some kind.
type	If type="inconsistency", an inconsistency model will be built. By default, type="consistency" and a consistency model is built. will be built.
effects	A string indicating the type of treatment effect relative to baseline. Options are "fixed" or "random".
scale	A string indicating the scale of the data, such as "Mean Difference" or "Log-Odds Ratio",
prior.mu	A string of BUGS code that defines priors on the baseline treatment effects. By default, independent normal priors are used with mean 0 and standard deviation 15u, where u is the largest maximum likelihood estimator in single trials (see van Valkenhoef, G et al. 2012).
prior.d	A string of BUGS code that defines define priors on relative treatment effects. By default, independent normal priors are used with mean 0 and standard deviation 15u, where u is the largest maximum likelihood estimator in single trials (see van Valkenhoef, G et al. 2012).
prior.sigma	A string of BUGS code that defines the prior on the variance of relative treatment effects. By default, a uniform distribution with range 0 to u is used, where u is the largest maximum likelihood estimator in single trials (see van Valkenhoef, G et al. 2012).

Value

nma.model returns an object of class BUGSnetModel which is a list containing the following components:

bugs - A long character string containing BUGS code that will be run in jags.

data - The data used in the BUGS code.

scale - The scale of the outcome, based on the chosen family and link function examples are "Risk Ratio" (relative risk), "Odds Ratio", "Mean Difference", "Hazard Ratio"

trt.key - Treatments mapped to integer numbers, used to run BUGS code.

...

References

van Valkenhoef, G, Lu, G, de Brock, B, Hillege, H, Ades, AE, Welton, NJ (2012). "Automating network meta-analysis." *Research Synthesis Methods*, **3**(4), 285-299.

Dias S, Sutton AJ, Welton NJ, Ades AE (2011). "NICE DSU technical support document 3. Heterogeneity: subgroups, meta-regression, bias and bias-adjustment." National Institute for Health and Clinical Excellence. Last updated April 2012.

See Also

data.prep, nma.run

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

#Random effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.re.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "random",
       type="consistency",
       time="followup"
       )

#Fixed effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.fe.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "fixed",
       type="consistency",
       time="followup"
       )
```

---

nma.model.shared *Create Bugs Model with Shared Parameters*

Description

Creates BUGS code which can be run through nma.run(). Handles a combination of arm-level and contrast-level data.

Usage

```
nma.model.shared(
  data_arm = NULL,
  data_contrast = NULL,
  outcome,
  differences,
  N = NULL,
  sd.a = NULL,
  se.diffs = NULL,
  var.arm1 = NULL,
  reference,
  type = "consistency",
  time = NULL,
  family = NULL,
  link = NULL,
  effects,
  prior.mu = "DEFAULT",
  prior.d = "DEFAULT",
  prior.sigma = "DEFAULT"
)
```

Arguments

data_arm	A BUGSnetData object containing the data from arm-based trials produced by data.prep()
data_contrast	A BUGSnetData object containing the data from contrast-based trials produced by data.prep()
outcome	A string indicating the name of your outcome variable for arm-based studies.
differences	A string indicating the name of the differences (outcome) for contrast-based studies
N	A string indicating the name of the variable containing the number of participants in each arm for arm-based data
sd.a	A string (only required for continuous outcomes with arm-level data) indicating variable name of the standard deviation of the outcome. Standard errors should be converted to standard deviation by multiplying by the square root of the sample size prior to using this function.
se.diffs	A string indicating the variable name of the standard errors of the differences in data_contrast.
var.arm1	A string (only required for contrast-based continuous data in networks with multi-arm trials) indicating the variable name of the variance of the treatment in arm 1 in each study
reference	A string for the treatment that will be seen as the 'referent' comparator and labeled as treatment 1 in the BUGS code. This is often a placebo or control drug of some kind.
type	If type="inconsistency", an inconsistency model will be built. By default, type="consistency" and a consistency model is built. will be built.
time	A string (only required for binomial-cloglog or poisson-log models) indicating the name of variable indicating person-time followup (e.g person years) or study followup time.

family	A string indicating the family of the distribution of the outcome for arm-based trials. Options are: "binomial", "normal", "poisson"
link	The link function for the nma model for arm-based models. Options are "logit" (binomial family), "log" (binomial family), "cloglog" (poisson family), "identity" (normal family).
effects	A string indicating the type of treatment effect relative to baseline. Options are "fixed" or "random".
prior.mu	A string of BUGS code that defines priors on the baseline treatment effects. By default, independent normal priors are used with mean 0 and standard deviation 15u, where u is the largest maximum likelihood estimator in single trials (see van Valkenhoef, G et al. 2012).
prior.d	A string of BUGS code that defines define priors on relative treatment effects. By default, independent normal priors are used with mean 0 and standard deviation 15u, where u is the largest maximum likelihood estimator in single trials (see van Valkenhoef, G et al. 2012).
prior.sigma	A string of BUGS code that defines the prior on the variance of relative treatment effects. By default, a uniform distribution with range 0 to u is used, where u is the largest maximum likelihood estimator in single trials (see van Valkenhoef, G et al. 2012).

Value

`nma.model` returns an object of class `BUGSnetModel` which is a list containing the following components:

`bugs` - A long character string containing BUGS code that will be run in `jags`.

`data` - The data used in the BUGS code.

`scale` - The scale of the outcome, based on the chosen family and link function examples are "Risk Ratio" (relative risk), "Odds Ratio", "Mean Difference", "Hazard Ratio"

`trt.key` - Treatments mapped to integer numbers, used to run BUGS code.

...

References

van Valkenhoef, G, Lu, G, de Brock, B, Hillege, H, Ades, AE, Welton, NJ (2012). "Automating network meta-analysis." *Research Synthesis Methods*, **3**(4), 285-299.

Dias S, Sutton AJ, Welton NJ, Ades AE (2011). "NICE DSU technical support document 3. Heterogeneity: subgroups, meta-regression, bias and bias-adjustment." National Institute for Health and Clinical Excellence. Last updated April 2012.

See Also

`data.prep`, `nma.run`

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
```

```
varname.s = "Study")

#Random effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.re.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "random",
       type="consistency",
       time="followup"
       )

#Fixed effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.fe.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "fixed",
       type="consistency",
       time="followup"
       )
```

`nma.rank` *Table and Plots of Treatment Rankings*

Description

Produces a SUCRA (Surface Under the Cumulative Ranking Curve) plot and table. A Sucra table summarizes the probabilities that each treatment is the best, second best...worst treatment in the network.

Usage

```
nma.rank(
  nma,
  largerbetter,
  sucra.lwd = 1,
  sucra.palette = "Set1",
  ranko.palette = "Blues",
  cov.value = NULL
)
```

Arguments

`nma`	A `BUGSnetRun` object produced by running `nma.run()`.

largerbetter	A boolean variable indicating whether a larger probability should indicate a more effective treatment (TRUE) or if a smaller probability should indicate a more effective treatment (FALSE).
sucra.lwd	Line width relative to the default (default=1) in the SUCRA plot.
sucra.palette	A string indicating the colour set from RcolorBrewer for the SUCRA plot. "Set1" is used by default and is automatically extended if there are many treatments in your network.
ranko.palette	A string indicating the colour set from RcolorBrewer for the rankogram. "Blues" is used by default and is automatically extended if there are many treatments in your network.
cov.value	Must be specified for meta-regression. This is the value of the covariate for which to report the results.

Value

ranktable - A rank table showing the probability of each treatment being the nth best treatment.

sucratable - A table showing the probability of each treatment being the nth best treatment or better and an overall SUCRA value for each treatment.

order - A vector containing the order of efficacy of treatments (from best to worst) based on their SUCRA value. This vector is useful for ordering treatments when creating the league heat plot with nma.league().

longtable - A long form table of ranking probabilities and SUCRA value.

sucraplot - A SUCRA plot showing the probability of each treatment being the nth best treatment or better.

rankogram - A rankogram showing the probability of each treatment being the nth best treatment.

See Also

nma.run, nma.league, nma.forest

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

#Random effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.re.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "random",
       type="consistency",
       time="followup"
       )

diabetes.re.c.res <- nma.run(diabetes.re.c,
```

```
n.adapt=100,
n.burnin=0,
n.iter=100)

#get sucra results
sucra_results <- nma.rank(nma = diabetes.re.c.res, largerbetter = FALSE)

#plot sucra results
sucra_results$sucraplot
```

`nma.regplot` *Plot of relative treatment effects vs covariate values for meta-regression.*

Description

Produces a plot of relative treatment effects on the linear scale vs covariate values for meta-regression.

Usage

```
nma.regplot(nma, x.range = NULL, lwd = 1, palette = "Set1")
```

Arguments

`nma`	A `BUGSnetRun` object produced by running `nma.run()`.
`x.range`	The range of the x axis (covariate values). By default, the range will be the same as in the data.
`lwd`	Line width relative to the default (default=1).
`palette`	A string indicating the colour set from RcolorBrewer for the plot. "set1" is great, but you may need a different colour set if there are many treatments in your network.

Value

`regplot` - A plot of the relative treatment effects vs covariate values for meta-regression.

See Also

`nma.run`

Examples

```
data(afib)

afib.slr <- data.prep(arm.data = afib,
varname.t = "treatment",
varname.s = "study")

#Random effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
afib.re.c <- nma.model(data=afib.slr,
outcome="events",
```

```
N="sampleSize",
reference="02",
family="binomial",
link="logit",
effects="random",
covariate="stroke",
prior.beta="EXCHANGEABLE")

afib.re.c.res <- nma.run(afib.re.c,
n.adapt=100,
n.burnin=0,
n.iter=100)

nma.regplot(afib.re.c.res)
```

`nma.run` *Run NMA model*

Description

Takes bugs code from an object produced by `nma.model` and runs model using `jags`.

Usage

```
nma.run(
  model,
  monitor = "DEFAULT",
  DIC = TRUE,
  n.adapt = 1000,
  n.burnin = floor(n.iter/2),
  n.iter,
  thin = 1,
  n.chains = 3,
  inits = "DEFAULT"
)
```

Arguments

`model`	A `BUGSnetModel` object produced by running `nma.model`.
`monitor`	A vector of all variables that you would like to monitor. Default is "DEFAULT" which will monitor the relative treatment effects `d` as well as `sigma` when a random effects model is used and the regression coefficients `beta` when meta-regression is used.
`DIC`	Default is TRUE and nodes required to calculate the DIC and other fit statistics are monitored. Otherwise you may set it to FALSE.
`n.adapt`	Number of adaptations for the mcmc chains.
`n.burnin`	Number of burnin iterations for the mcmc chains.
`n.iter`	Number of iterations for the mcmc chains.
`thin`	Thinning factor for the mcmc chains. Default is 1.
`n.chains`	Number of mcmc chains. Default is 3.

`inits` Specifies initial values and random number generator options for each chain. The "DEFAULT" option uses the R random seed to set the JAGS random seeds. Non-default options are passed directly to `jags.model`. In order to use the JAGS default initialization, set inits to NULL. See `jags.model` for more info.

Value

`nma.run` returns an object of class `BUGSnetRun` which is a list containing the following components:

`samples` - The MCMC samples produced by running the BUGS model.

`model` - The `BUGSnetModel` object obtained from `nma.model` which was used to run `jags`.

`scale` - The scale of the outcome, based on the chosen family and link function.

`trt.key` - Treatments mapped to numbers, used to run BUGS code.

`family` - Family that was used for the NMA model (e.g normal, binomial, poisson)

`link` - Link function that was used for the NMA model (e.g normal, binomial, poisson)

See Also

`nma.model`, `nma.fit`, `nma.league`, `nma.rank`, `nma.forest`, `nma.regplot`, `nma.trace`, `jags.model`

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

#Random effects, consistency model.
#Binomial family, cloglog link. This implies that the scale will be the Hazard Ratio.
diabetes.re.c <- nma.model(data = diabetes.slr,
       outcome = "diabetes",
       N = "n",
       reference = "Placebo",
       family = "binomial",
       link = "cloglog",
       effects = "random",
       type="consistency",
       time="followup"
       )

diabetes.re.c.res <- nma.run(diabetes.re.c,
n.adapt=100,
n.burnin=0,
n.iter=100)
```

nma.trace *Deprecated. Please use* nma.diag *instead.*

Description

Deprecated. Produces traceplots of the MCMC chains obtained from nma.run()

Usage

```
nma.trace(nma, n = "all", thin = 1, colours = "DEFAULT")
```

Arguments

nma	A BUGSnetRun object produced by running nma.run().
n	Integer which limits the number of printed variables to the first n. Default is "all" which plots every variable.
thin	Thinning factor for the mcmc chains. Default is 1.
colours	An optional vector of colors, one for each chain.

See Also

nma.diag

parkinsons *Parkinsons contrast data*

Description

This dataset was used to demonstrate network meta-analysis in the NICE-DSU TSD 2 (example 7). It includes information on seven clinical trials examining the effects of five different treatments for Parkinsons on the mean off-time reduction (continuous outcome). This dataset is in the form of mean differences in order to demonstrate the contrast data.

Usage

```
data(parkinsons)
```

Format

A dataset.

References

Dias S, Welton NJ, Sutton AJ, Ades AE (2011). "NICE DSU technical support document 2: a generalised linear modelling framework for pairwise and network meta-analysis of randomised controlled trials." National Institute for Health and Clinical Excellence.

Examples

```
data(parkinsons)
```

parkinsons_arm *Parkinsons arm-based data*

Description

This dataset was used to demonstrate network meta-analysis in the NICE-DSU TSD 2 (examples 5 and 8). It includes arm-based information on three out of a total of seven clinical trials examining the effects of five different treatments for Parkinsons on the mean off-time reduction (continuous outcome). This dataset can be used in combination with rows 9 to 15 of the `parkinsons` dataset to demonstrate the use of shared parameter models in BUGSnet.

Usage

```
data(parkinsons_arm)
```

Format

A dataset.

References

Dias S, Welton NJ, Sutton AJ, Ades AE (2011). "NICE DSU technical support document 2: a generalised linear modelling framework for pairwise and network meta-analysis of randomised controlled trials." National Institute for Health and Clinical Excellence.

Examples

```
data(parkinsons_arm)
```

pma *Pairwise meta-analysis*

Description

implements pairwise meta-analysis via the package `meta`

Usage

```
pma(
  data,
  name.trt1,
  name.trt2,
  outcome,
  N,
  sd = NULL,
  time = NULL,
  type.outcome,
  method = "MH",
  method.tau = "DL",
  sm
)
```

Arguments

data	A BUGSnetData object produced by data.prep()
name.trt1	A string indicating the name of the comparator treatment (often Placebo)
name.trt2	A string indicating the name of the experimental treatment
outcome	A string indicating the name of your outcome variable
N	A string indicating the name of the variable containing the number of participants in each arm
sd	A string (only required when type.outcome="continuous") indicating variable name of the standard deviation of the outcome
time	A string required when type.outcome = "rate". Name of variable indicating person-time followup (e.g person years).
type.outcome	A string. Options are: "binomial", "continuous", "rate" (e.g # of events and # person-years reported)
method	A character string indicating what type of test was used. For more info, see meta's documentation.
method.tau	A character string indicating which method is used to estimate the between study variance. Either "DL", "PM", "REML", "ML", "HS", "SJ", "HE", or "EB", can be abbreviated.
sm	A character string indicating which summary measure ("RR", "OR", "RD", or ASD") is to be used for pooling of studies.

Value

A forest plot as produced by the package meta

raw - dataset containing summary statistics of meta-analysis (effect estimates, confidence bounds, I-squared, Q-statistic)

Note

Depending on the the value of type.outcome, this function will implement the functions metabin (dichotomous outcomes), metacont (continuos outcomes), or metainc (rate outcomes) from the package meta.

See Also

data.prep

Examples

```
data(diabetes.sim)

diabetes.slr <- data.prep(arm.data = diabetes.sim,
varname.t = "Treatment",
varname.s = "Study")

pma(data = diabetes.slr,
type.outcome="binomial",
sm="OR",
name.trt1 = "Placebo",
name.trt2 = "Diuretic",
```

```
outcome = "diabetes",
N = "n")
```

`thrombolytic` *Thrombolytic data*

Description

The thrombolytic drugs dataset compiled by Boland et al. (2003) has been used to demonstrate indirect-treatment comparison methods within previous publications and other R packages such as 'gemtc'. The dataset consists of information on a binary outcome (mortality within 30 or 35 days) and eight treatments (eight different thrombolytic drugs used after a patient has had an acute myocardial infarction) extracted from 28 clinical trials (including two, three-armed trials).

Usage

```
data(thrombolytic)
```

Format

A dataset.

References

Boland A, Dundar Y, Bagust A, Haycox A, Hill R, Mota RM, Walley T, A D (2003). "Early thrombolysis for the treatment of acute myocardial infarction: a systematic review and economic evaluation." *Health Technology Assessment*, **7**(15), 1–136.

Dias S, Welton NJ, Caldwell DM, AE A (2010). "Checking consistency in mixed treatment comparison meta-analysis." *Statistics in medicine*, **7-8**(29), 932–44.

Lu G, Ades AE (2006). "Assessing evidence inconsistency in mixed treatment comparisons." *Journal of the American Statistical Association*, **101**(474), 447–459.

van Valkenhoef, G, Lu, G, de Brock, B, Hillege, H, Ades, AE, Welton, NJ (2012). "Automating network meta-analysis." *Research Synthesis Methods*, **3**(4), 285-299.

Examples

```
data(thrombolytic)
```

Index