Package 'MBNMAdose'

September 10, 2020

Type Package

Title Dose-Response MBNMA Models

Version 0.3.0

Language en-GB

Date 2020-09-10

URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4999602/

Maintainer Hugo Pedder <hugopedder@gmail.com>

Description Fits Bayesian dose-response model-based network meta-analysis (MBNMA) that incorporate multiple doses within an agent by modelling different dose-response functions, as described by Mawdsley et al. (2016) <doi:10.1002/psp4.12091>.
By modelling dose-response relationships this can connect networks of evidence that might otherwise be disconnected, and can improve precision on treatment estimates. Several common dose-response functions are provided; others may be added by the user. Various characteristics and assumptions can be flexibly added to the models, such as shared class effects. The consistency of direct and indirect evidence in the network can be assessed using unrelated mean effects models and/or by node-splitting at the treatment level.

License GPL-3

Depends R (>= 3.0.2)

Imports grDevices, stats, graphics, utils, scales, utf8, formatR, dplyr (>= 0.7.4), R2jags (>= 0.5-7), rjags (>= 4-8), magrittr (>= 1.5), checkmate (>= 1.8.5), Rdpack (>= 0.11-0), igraph (>= 1.1.2), rgeos (>= 0.5-2), reshape2 (>= 1.4.3)

Suggests overlapping (>= 1.5.0), ggplot2 (>= 2.2.1), RColorBrewer (>= 1.1-2), mcmcplots (>= 0.4.3), testthat (>= 1.0.2), crayon (>= 1.3.4), Hmisc (>= 4.4-1), knitr, rmarkdown

SystemRequirements JAGS (>= 4.3.0) (https://mcmc-jags.sourceforge.net/)

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.1

RdMacros Rdpack

NeedsCompilation no

Author Hugo Pedder [aut, cre],
Lujin Li [rev],
Adil Karim [ctb]

Repository CRAN

Date/Publication 2020-09-10 16:40:07 UTC

R topics documented:

`add_index` *Add arm indices and agent identifiers to a dataset*

Description

Adds arm indices (`arms`, `narms`) to a dataset and adds numeric identifiers for agent and class (if included in the data).

Usage

```
add_index(data.ab, agents = NULL, treatments = NULL)
```

Arguments

`data.ab` A data frame of arm-level data in "long" format containing the columns:

- `studyID` Study identifiers
- `dose` Numeric data indicating the dose (must take positive values)
- `agent` Agent identifiers (can be numeric, factor or character)
- `y` Numeric data indicating the aggregate response for a continuous outcome. Required for continuous data.
- `se` Numeric data indicating the standard error for a given observation. Required for continuous data.
- `r` Numeric data indicating the number of responders within a study arm. Required for binomial or poisson data.
- `N` Numeric data indicating the total number of participants within a study arm. Required for binomial data
- `E` Numeric data indicating the total exposure time for participants within a study arm. Required for poisson data.

- `class` An optional column indicating a particular class code. Agents with the same identifier must also have the same class code.

`agents` A character string of agent names used to force a particular agent ordering. Default is `NULL`, which automatically orders `Placebo` (dose=0) as agent `1` and then subsequent agents by the order given in `data.ab`

`treatments` A character string of treatment names used to force a particular treatment ordering. Default is `NULL`, which automatically orders `Placebo` (dose=0) as treatment `1` and then subsequent treatments by the order of agents and doses (smallest to highest) given in `data.ab`

Value

A data frame similar to `data.ab` but with additional columns:

- `arm` Arm identifiers coded for each study
- `narm` The total number of arms in each study

If `agent` or `class` are non-numeric or non-sequential (i.e. with missing numeric codes), agents/classes in the returned data frame will be numbered and recoded to enforce sequential numbering (a warning will be shown stating this).

`alog_pcfb` *Studies of alogliptin for lowering blood glucose concentration in patients with type II diabetes*

Description

A dataset from a systematic review of Randomised-Controlled Trials (RCTs) comparing different doses of alogliptin with placebo (Langford et al. 2016). The systematic review was simply performed and was intended to provide data to illustrate a statistical methodology rather than for clinical inference. Alogliptin is a treatment aimed at reducing blood glucose concentration in type II diabetes. The outcome is continuous, and aggregate data responses correspond to the mean change in HbA1c from baseline to follow-up in studies of at least 12 weeks follow-up. The dataset includes 14 Randomised-Controlled Trials (RCTs), comparing 5 different doses of alogliptin with placebo, leading to 6 different treatments (combination of dose and agent) within the network.

A dataset from a systematic review of Randomised-Controlled Trials (RCTs) comparing different biologics for the treatment of psoriasis (Warren et al. 2019). The systematic review was simply performed and was intended to provide data to illustrate a statistical methodology rather than for clinical inference. Alogliptin is a treatment aimed at reducing blood glucose concentration in type II diabetes. The outcome is continuous, and aggregate data responses correspond to the mean change in HbA1c from baseline to follow-up in studies of at least 12 weeks follow-up. The dataset includes 14 Randomised-Controlled Trials (RCTs), comparing 5 different doses of alogliptin with placebo, leading to 6 different treatments (combination of dose and agent) within the network.

Usage

```
alog_pcfb

alog_pcfb
```

Format

A data frame in long format (one row per arm and study), with 46 rows and 6 variables:

- `studyID` Study identifiers
- `agent` Character data indicating the agent to which participants were randomised
- `dose` Numeric data indicating the standardised dose received
- `y` Numeric data indicating the mean change from baseline in blood glucose concentration (mg/dL) in a study arm
- `se` Numeric data indicating the standard error for the mean change from baseline in blood glucose concentration (mg/dL) in a study arm

A data frame in long format (one row per arm and study), with 46 rows and 6 variables:

- `studyID` Study identifiers
- `agent` Character data indicating the agent to which participants were randomised
- `dose` Numeric data indicating the standardised dose received
- `y` Numeric data indicating the mean change from baseline in blood glucose concentration (mg/dL) in a study arm
- `se` Numeric data indicating the standard error for the mean change from baseline in blood glucose concentration (mg/dL) in a study arm

Details

`alog_pcfb` is a data frame in long format (one row per arm and study), with the variables `studyID`, `agent`, `dose`, `y`, `se`, and `N`.

`alog_pcfb` is a data frame in long format (one row per arm and study), with the variables `studyID`, `agent`, `dose`, `y`, `se`, and `N`.

References

Langford O, Aronson JK, van Valkenhoef G, Stevens RJ (2016). "Methods for meta-analysis of pharmacodynamic dose-response data with application to multi-arm studies of alogliptin." *Stat Methods Med Res*. ISSN 1477-0334 (Electronic) 0962-2802 (Linking), doi: 10.1177/0962280216637093, https://pubmed.ncbi.nlm.nih.gov/26994216/.

Warren RB, Gooderham M, Burge R, Zhu B, Amato D, Liu KH, Shrom D, Guo J, Brnabic A, Blauvelt A (2019). "Comparison of cumulative clinical benefits of biologics for the treatment of psoriasis over 16 weeks: Results from a network meta-analysis." *J Am Acad Dermatol*, **82**(5), 1138-1149.

Langford O, Aronson JK, van Valkenhoef G, Stevens RJ (2016). "Methods for meta-analysis of pharmacodynamic dose-response data with application to multi-arm studies of alogliptin." *Stat*

Methods Med Res. ISSN 1477-0334 (Electronic) 0962-2802 (Linking), doi: 10.1177/0962280216637093, https://pubmed.ncbi.nlm.nih.gov/26994216/.

Warren RB, Gooderham M, Burge R, Zhu B, Amato D, Liu KH, Shrom D, Guo J, Brnabic A, Blauvelt A (2019). "Comparison of cumulative clinical benefits of biologics for the treatment of psoriasis over 16 weeks: Results from a network meta-analysis." *J Am Acad Dermatol*, **82**(5), 1138-1149.

`check.likelink` *Check likelihood and link function*

Description

Checks that likelihood and link function is provided and confirm that the correct form of data is provided.

Usage

```
check.likelink(data.ab, likelihood = NULL, link = NULL)
```

Arguments

`data.ab`	A data frame of arm-level data in "long" format containing the columns: • `studyID` Study identifiers • `dose` Numeric data indicating the dose (must take positive values) • `agent` Agent identifiers (can be numeric, factor or character) • `y` Numeric data indicating the aggregate response for a continuous outcome. Required for continuous data. • `se` Numeric data indicating the standard error for a given observation. Required for continuous data. • `r` Numeric data indicating the number of responders within a study arm. Required for binomial or poisson data. • `N` Numeric data indicating the total number of participants within a study arm. Required for binomial data • `E` Numeric data indicating the total exposure time for participants within a study arm. Required for poisson data. • `class` An optional column indicating a particular class code. Agents with the same identifier must also have the same class code.
`likelihood`	A string indicating the likelihood to use in the model. Can take either `"binomial"`, `"normal"` or `"poisson"`. If left as `NULL` the likelihood will be inferred from the data.
`link`	A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. `"logit"`, `"log"`, `"probit"`, `"cloglog"`) or be assigned the value `"identity"` for and identity link function. If left as `NULL` the link function will be automatically assigned based on the likelihood.

cumrank *Plot cumulative ranking curves from MBNMA models*

Description

Plot cumulative ranking curves from MBNMA models

Usage

```
cumrank(x, params = NULL, sucra = TRUE, ...)
```

Arguments

x	An object of class "mbnma.rank" generated by rank.mbnma()
params	A character vector of named parameters in the model that vary by either agent or class (depending on the value assigned to level). If left as NULL (the default), then ranking will be calculated for all available parameters that vary by agent/class.
sucra	A logical object to indicate whether Surface Under Cumulative Ranking Curve (SUCRA) values should be calculated and returned as a data frame. Areas calculated using readWKT.
...	Arguments to be sent to ggplot::geom_line()

Value

Line plots showing the cumulative ranking probabilities for each agent/class and dose-response parameter in x. The object returned is a list which contains the plot (an object of class(c("gg", "ggplot")) and a data frame of SUCRA values if sucra = TRUE.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Estimate rankings  from an Emax dose-response MBNMA
emax <- mbnma.emax(network, emax="rel", ed50="rel", method="random")
ranks <- rank(emax)

# Plot cumulative rankings for both dose-response parameters simultaneously
# Note that SUCRA values are also returned
cumrank(ranks)
```

devplot *Plot deviance contributions from an MBNMA model*

Description

Plot deviance contributions from an MBNMA model

Usage

```
devplot(
  mbnma,
  plot.type = "scatter",
  facet = TRUE,
  dev.type = "resdev",
  n.iter = mbnma$BUGSoutput$n.iter,
  n.thin = mbnma$BUGSoutput$n.thin,
  ...
)
```

Arguments

mbnma	An S3 object of class "mbnma" generated by running a dose-response MBNMA model
plot.type	Deviances can be plotted either as scatter points ("scatter" - the default) or as boxplots ("box")
facet	A boolean object that indicates whether or not to facet (by agent for MBNMAdose and by treatment for MBNMAtime)
dev.type	*STILL IN DEVELOPMENT FOR MBNMAdose!* Deviances to plot - can be either residual deviances ("resdev", the default) or deviances ("dev")
n.iter	number of total iterations per chain (including burn in; default: 2000)
n.thin	thinning rate. Must be a positive integer. Set n.thin > 1 to save memory and computation time if n.iter is large. Default is max(1,floor(n.chains * (n.iter-n.burnin) / 1000)) which will only thin if there are at least 2000 simulations.
...	Arguments to be sent to ggplot2::geom_point() or ggplot2::geom_boxplot

Details

Deviances should only be plotted for models that have converged successfully. If deviance contributions have not been monitored in mbnma$parameters.to.save then additional iterations will have to be run to get results for these.

For MBNMAtime, deviance contributions cannot be calculated for models with a multivariate likelihood (i.e. those that account for correlation between observations) because the covariance matrix in these models is treated as unknown (if rho = "estimate") and deviance contributions will be correlated.

Value

Generates a plot of deviance contributions and returns a list containing the plot (as an object of `class(c("gg","ggplot"))`), and a data.frame of posterior mean deviance/residual deviance contributions for each observation.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Run an Emax dose-response MBNMA and predict responses
emax <- mbnma.emax(network, method="random")

# Plot deviances
devplot(emax)

# Plot deviances using boxplots
devplot(emax, plot.type="box")

# Plot deviances on a single plot (not facetted by agent)
devplot(emax, facet=FALSE, plot.type="box")

# A data frame of deviance contributions can be obtained from the object
#returned by `devplot`
devs <- devplot(emax)
head(devs$dev.data)

# Other deviance contributions not currently implemented but in future
#it will be possible to plot them like so
#devplot(emax, dev.type="dev")
```

---

drop.comp *Drop treatments from multi-arm (>2) studies for node-splitting*

Description

Drops arms in a way which preserves connectivity and equally removes data from each treatment in a nodesplit comparison (so as to maximise precision)

Usage

```
drop.comp(ind.df, drops, comp, start = 1)
```

Arguments

ind.df	A data frame in long format (one arm per row) from which to drop treatments
drops	A vector of study identifiers from which to drop treatments
comp	A numeric vector of length 2 that contains treatment codes corresponding to the comparison for node-splitting
start	Can take either 0 or 1 to indicate whether to drop the treatment in comp[1] (0) or comp[2] (1)

---

drop.disconnected *Drop studies that are not connected to the network reference treatment*

Description

Drop studies that are not connected to the network reference treatment

Usage

```
drop.disconnected(network, connect.dose = FALSE)
```

Arguments

network	An object of class mbnma.network.
connect.dose	A boolean object to indicate whether treatments should be kept in the network if they connect via the simplest possible dose-response relationship (TRUE) or not (FALSE). Simplest possible dose-response relationship is any function with a single dose-response parameter (e.g. linear, exponential)

Value

A list containing a single row per arm data frame containing only studies that are connected to the network reference treatment, and a character vector of treatment labels

Examples

```
# Using the triptans headache dataset
network <- mbnma.network(HF2PPITT)
drops <- drop.disconnected(network)

# No studies have been dropped since network is fully connected
length(unique(network$data.ab$studyID))==length(unique(drops$data.ab$studyID))


# Make data with no placebo
noplac.df <- network$data.ab[network$data.ab$narm>2 & network$data.ab$agent!=1,]
net.noplac <- mbnma.network(noplac.df)

# Studies are dropped as some only connect via the dose-response function
```

```
drops <- drop.disconnected(net.noplac, connect.dose=FALSE)
length(unique(net.noplac$data.ab$studyID))==length(unique(drops$data.ab$studyID))

# Studies are not dropped if they connect via the dose-response function
drops <- drop.disconnected(net.noplac, connect.dose=TRUE)
length(unique(net.noplac$data.ab$studyID))==length(unique(drops$data.ab$studyID))
```

`fitplot` *Plot fitted values from MBNMA model*

Description

Plot fitted values from MBNMA model

Usage

```
fitplot(
  mbnma,
  disp.obs = TRUE,
  n.iter = mbnma$BUGSoutput$n.iter,
  n.thin = mbnma$BUGSoutput$n.thin,
  ...
)
```

Arguments

`mbnma`	An S3 object of class `"mbnma"` generated by running a dose-response MBNMA model
`disp.obs`	A boolean object to indicate whether raw data responses should be plotted as points on the graph
`n.iter`	number of total iterations per chain (including burn in; default: 2000)
`n.thin`	thinning rate. Must be a positive integer. Set `n.thin > 1` to save memory and computation time if `n.iter` is large. Default is `max(1,floor(n.chains * (n.iter-n.burnin) / 1000))` which will only thin if there are at least 2000 simulations.
`...`	Arguments to be sent to `ggplot2::geom_point()` or `ggplot2::geom_line()`

Details

Fitted values should only be plotted for models that have converged successfully. If fitted values (`theta`) have not been monitored in `mbnma$parameters.to.save` then additional iterations will have to be run to get results for these.

Value

Generates a plot of fitted values from the MBNMA model and returns a list containing the plot (as an object of `class(c("gg","ggplot"))`), and a data.frame of posterior mean fitted values for each observation.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Run an Emax dose-response MBNMA and predict responses
emax <- mbnma.emax(network, method="random")

# Plot fitted values and observed values
fitplot(emax)

# Plot fitted values only
fitplot(emax, disp.obs=FALSE)

# A data frame of fitted values can be obtained from the object
#returned by `fitplot`
fits <- fitplot(emax)
head(fits$fv)
```

---

genspline *Generates spline basis matrices for fitting to dose-response function*

Description

Generates spline basis matrices for fitting to dose-response function

Usage

```
genspline(x, spline = "rcs", knots = 3, ord = 4, max.dose = max(x))
```

Arguments

`x`	A numeric vector indicating all doses of an agent available in the dataset (including placebo)
`spline`	Indicates the type of spline function. Can be either natural cubic spline (`"ns"`), restricted cubic spline (`"rcs"`) or B-spline (`"bs"`).
`knots`	The number/location of knots if a restricted cubic spline dose-response function is fitted (`fun="rcs"`). If a single number is given it indicates the number of knots (they will be equally spaced across the range of doses). If a numeric vector is given it indicates the location of the knots. Minimum number of knots is 3.

ord	a positive integer giving the order of the spline function. This is the number of coefficients in each piecewise polynomial segment, thus a cubic spline has order 4. Defaults to 4.
max.dose	A number indicating the maximum dose between which to calculate knot points.

`get.prior` *Get current priors from JAGS model code*

Description

Identical to `get.prior()` in `MBNMAtime` package. This function takes JAGS model presented as a string and identifies what prior values have been used for calculation.

Usage

```
get.prior(model)
```

Arguments

model	A character object of JAGS MBNMA model code

Details

Even if an MBNMA model that has not initialised successfully and results have not been calculated, the JAGS model for it is saved in `mbnma$model.arg$jagscode` and therefore priors can still be obtained. This allows for priors to be changed even in failing models, which may help solve issues with compiling or updating.

Value

A character vector, each element of which is a line of JAGS code corresponding to a prior in the JAGS code.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Run an Emax dose-response MBNMA
result <- mbnma.emax(network, emax="rel", ed50="rel", method="random")

# Obtain model prior values
print(result$model.arg$priors)

# Priors when using mbnma.run with an exponential function
result <- mbnma.run(network, fun="exponential", beta.1="rel", method="random")
print(result$model.arg$priors)
```

`get.relative` *Calculates relative effects between treatments in an MBNMA model*

Description

Calculates relative effects between treatments in an MBNMA model

Usage

```
get.relative(mbnma, treatments = list())
```

Arguments

`mbnma`	An object of `class("mbnma")`
`treatments`	A list whose elements each represent different treatments. Treatment is defined as a combination of agent and dose. Only agents specified in `mbnma` can be included. Each element in `treatments` is named corresponding to the agent and contains a numeric vector of doses. Relative effects will be calculated between all treatments specified in `treatments`.

Value

An array of length(treatments) x length(treatments) x nsims, where `nsims` is the number of iterations monitored in `mbnma`. The array contains the individual MCMC values for each relative effect calculated between all `treatments`. The direction of effect is for the row-defined treatment versus the column-defined treatment.

Examples

```
# Using the osteoarthritis data
network <- mbnma.network(osteopain_2wkabs)

expon <- mbnma.exponential(network, method="random")

# Calculate relative effects between:
# Celebrex 100mg/d, Celebrex 200mg/d, Tramadol 100mg/d
rel.eff <- get.relative(expon, treatments=list("Celebrex"=c(100,200), "Tramadol"=100))
```

getjagsdata *Prepares data for JAGS*

Description

Converts MBNMA data frame to a list for use in JAGS model

Usage

```
getjagsdata(
  data.ab,
  class = FALSE,
  likelihood = "binomial",
  link = "logit",
  level = "agent",
  fun = NULL,
  nodesplit = NULL,
  knots = 3
)
```

Arguments

data.ab	A data frame of arm-level data in "long" format containing the columns: • studyID Study identifiers • dose Numeric data indicating the dose (must take positive values) • agent Agent identifiers (can be numeric, factor or character) • y Numeric data indicating the aggregate response for a continuous outcome. Required for continuous data. • se Numeric data indicating the standard error for a given observation. Required for continuous data. • r Numeric data indicating the number of responders within a study arm. Required for binomial or poisson data. • N Numeric data indicating the total number of participants within a study arm. Required for binomial data • E Numeric data indicating the total exposure time for participants within a study arm. Required for poisson data. • class An optional column indicating a particular class code. Agents with the same identifier must also have the same class code.
class	A boolean object indicating whether or not data.ab contains information on different classes of treatments
likelihood	A string indicating the likelihood to use in the model. Can take either "binomial", "normal" or "poisson". If left as NULL the likelihood will be inferred from the data.

link	A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. "logit", "log", "probit", "cloglog") or be assigned the value "identity" for and identity link function. If left as NULL the link function will be automatically assigned based on the likelihood.
level	Can take either "agent" to indicate that data should be at the agent- level (for MBNMA) or "treatment" to indicate that data should be at the treatment- level (for NMA)
fun	A character vector specifying a functional form to be assigned to the dose-response. Options are given in details.
nodesplit	A numeric vector of length 2 containing treatment codes on which to perform an MBNMA nodesplit.
knots	The number/location of knots if a restricted cubic spline dose-response function is fitted (fun="rcs"). If a single number is given it indicates the number of knots (they will be equally spaced across the range of doses). If a numeric vector is given it indicates the location of the knots. Minimum number of knots is 3.

Value

A named list of numbers, vector, matrices and arrays to be sent to JAGS. List elements are:

- If likelihood="normal":
  - y An array of mean responses for each arm within each study
  - se An array of standard errors for each arm within each study
- If likelihood="binomial":
  - r An array of the number of responses/count for each each arm within each study
  - N An array of the number of participants for each arm within each study
- If likelihood="poisson":
  - r An array of the number of responses/count for each each arm within each study
  - E An array of the total exposure time for each arm within each study
- dose A matrix of doses for each arm within each study (if level="agent")
- narm A numeric vector with the number of arms per study
- NS The total number of studies in the dataset
- Nagent The total number of agents in the dataset (if level="agent")
- agent A matrix of agent codes within each study (if level="agent")
- NT The total number of treatment in the dataset (if level="treatment")
- treatment A matrix of treatment codes within each study (if level="treatment")
- Nclass Optional. The total number of classes in the dataset
- class Optional. A matrix of class codes within each study
- classkey Optional. A vector of class codes that correspond to agent codes. Same length as the number of agent codes.
- split.ind Optional. A matrix indicating whether a specific arm contributes evidence to a nodesplit comparison.

Examples

```
# Using the triptans headache dataset
network <- mbnma.network(HF2PPITT)

jagsdat <- getjagsdata(network$data.ab, likelihood="binomial", link="logit")


# Get JAGS data with class
df <- HF2PPITT
df$class <- ifelse(df$agent=="placebo", "placebo", "active")
netclass <- mbnma.network(df)

jagsdat <- getjagsdata(netclass$data.ab, class=TRUE)


# Get JAGS data at the treatment level for Network Meta-Analysis
network <- mbnma.network(HF2PPITT)

jagsdat <- getjagsdata(network$data.ab, level="treatment")
```

GoutSUA_2wkCFB *Studies of treatments for Serum Uric Acid reduction in patients with gout*

Description

A dataset from a systematic review of interventions for lowering Serum Uric Acid (SUA) concentration in patients with gout **(not published previously)**. The outcome is continuous, and aggregate data responses correspond to the mean change from baseline in SUA in mg/dL at 2 weeks follow-up. The dataset includes 10 Randomised-Controlled Trials (RCTs), comparing 5 different agents, and placebo. Data for one agent (RDEA) arises from an RCT that is not placebo-controlled, and so is not connected to the network directly. In total there were 19 different treatments (combination of dose and agent).

Usage

```
GoutSUA_2wkCFB
```

Format

A data frame in long format (one row per arm and study), with 27 rows and 5 variables:

- `studyID` Study identifiers
- `y` Numeric data indicating the mean change from baseline in SUA in a study arm
- `se` Numeric data indicating the standard error for the mean change from baseline in SUA in a study arm
- `agent` Character data indicating the agent to which participants were randomised
- `dose` Numeric data indicating the standardised dose received

Source

Pfizer Ltd.

`HF2PPITT` *Studies of triptans for headache pain relief*

Description

A dataset from a systematic review of interventions for pain relief in migraine (Thorlund et al. 2014). The outcome is binary, and represents (as aggregate data) the proportion of participants who were headache-free at 2 hours. Data are from patients who had had at least one migraine attack, who were not lost to follow-up, and who did not violate the trial protocol. The dataset includes 70 Randomised-Controlled Trials (RCTs), comparing 7 triptans with placebo. Doses are standardised as relative to a "common" dose, and in total there are 23 different treatments (combination of dose and agent).

Usage

```
HF2PPITT
```

Format

A data frame in long format (one row per arm and study), with with 181 rows and 6 variables:

- `studyID` Study identifiers
- `AuthorYear` The author and year published of the study
- `N` Numeric data indicating the number of participants in a study arm
- `r` Numeric data indicating the number of responders (headache free at 2 hours) in a study arm
- `dose` Numeric data indicating the standardised dose received
- `agent` Factor data indicating the agent to which participants were randomised

Source

There are no references for Rd macro \insertAllCites on this help page.

`inconsistency.loops` *Identify comparisons in loops that fulfil criteria for node-splitting*

Description

Identify comparisons informed by both direct and indirect evidence from independent sources, which therefore fulfil the criteria for testing for inconsistency via node-splitting. Follows the method of van Valkenhoef et al. (2016).

Usage

```
inconsistency.loops(df, checkindirect = TRUE, incldr = FALSE)
```

Arguments

`df`	A data frame containing variables `studyID` and `treatment` (as numeric codes) that indicate which treatments are used in which studies. If `checkindirect = TRUE` then variables `agent` and `dose` are also required.
`checkindirect`	A boolean object to indicate whether or not to perform an additional check to ensure network remains connected even after dropping direct evidence on a comparison. Default is `TRUE` and should be kept as `TRUE` if working with dose-response data, though this requires further computational iterations to confirm. If set to `FALSE`, additional comparisons may be identified, though computation will be much more rapid.
`incldr`	A boolean object indicating whether or not to allow for indirect evidence contributions via the dose-response relationship. This can be used when node-splitting in dose-response MBNMA to allow for a greater number of potential loops in which to check for consistency.

Details

Similar to `gemtc::mtc.nodesplit.comparisons()` but uses a fixed reference treatment and therefore identifies fewer loops in which to test for inconsistency. Heterogeneity can also be parameterised as inconsistency and so testing for inconsistency in additional loops whilst changing the reference treatment would also be identifying heterogeneity. Depends on `igraph`.

Value

A data frame of comparisons that are informed by direct and indirect evidence from independent sources. Each row of the data frame is a different treatment comparison. Numerical codes in `t1` and `t2` correspond to treatment codes. `path` indicates the treatment codes that connect the shortest path of indirect evidence.

If `incldr=TRUE` then `path` may indicate `doseresp` for some comparisons. These are comparisons for which indirect evidence is only available via the dose-response relationship. The two numbers given after (e.g. 3 2) indicate the number of doses available in the indirect evidence with which to estimate the dose-response function for the treatments in `t1` and `t2` respectively/

References

van Valkenhoef G, Dias S, Ades AE, Welton NJ (2016). "Automated generation of node-splitting models for assessment of inconsistency in network meta-analysis." *Res Synth Methods*, **7**(1), 80-93. ISSN 1759-2887 (Electronic) 1759-2879 (Linking), doi: 10.1002/jrsm.1167, https://pubmed.ncbi.nlm.nih.gov/26461181/.

Examples

```
# Identify comparisons informed by direct and indirect evidence
#in triptans dataset
network <- mbnma.network(HF2PPITT)
inconsistency.loops(network$data.ab)

# Include indirect evidence via dose-response relationship
inconsistency.loops(network$data.ab, incldr=TRUE)



# Do not perform additional connectivity check on data
data <- data.frame(studyID=c(1,1,2,2,3,3,4,4,5,5,5),
            treatment=c(1,2,1,3,2,3,3,4,1,2,4)
            )
inconsistency.loops(data, checkindirect=FALSE)
```

`mbnma.comparisons` *Identify unique comparisons within a network*

Description

Identify unique contrasts within a network that make up all the head-to-head comparisons. Repetitions of the same treatment comparison are grouped together.

Usage

```
mbnma.comparisons(data)
```

Arguments

`data`	A data frame containing variables `studyID` and `treatment` (as numeric codes) that indicate which treatments are used in which studies.

Value

A data frame of unique comparisons in which each row represents a different comparison. `t1` and `t2` indicate the treatment codes that make up the comparison. `nr` indicates the number of times the given comparison is made within the network.

If there is only a single follow-up observation for each study within the dataset (i.e. as for standard network meta-analysis) `nr` will represent the number of studies that compare treatments `t1` and `t2`.

If there are multiple observations for each study within the dataset (as in time-course MBNMA) `nr` will represent the number of time points in the dataset in which treatments `t1` and `t2` are compared.

Examples

```
data <- data.frame(studyID=c(1,1,2,2,3,3,4,4,5,5,5),
  treatment=c(1,2,1,3,2,3,3,4,1,2,4)
  )

# Identify unique comparisons within the data
mbnma.comparisons(data)


# Using the triptans headache dataset
network <- mbnma.network(HF2PPITT) # Adds treatment identifiers
mbnma.comparisons(network$data.ab)
```

`mbnma.emax` *Run MBNMA model with an Emax dose-response function (without Hill parameter)*

Description

Fits a Bayesian model-based network meta-analysis (MBNMA) with a defined dose-response function. Follows the methods of Mawdsley et al. (2016). This function acts as a wrapper for `mbnma.run()` that uses more clearly defined parameter names.

Usage

```
mbnma.emax(
  network,
  emax = "rel",
  ed50 = "rel",
  method = "common",
  class.effect = list(),
  UME = FALSE,
  cor = TRUE,
  var.scale = NULL,
  parameters.to.save = NULL,
  pd = "pv",
  parallel = FALSE,
  likelihood = NULL,
  link = NULL,
  priors = NULL,
  arg.params = NULL,
```

```
  ...
)
```

Arguments

network	An object of class mbnma.network.
emax	Refers to the Emax parameter of the Emax dose-response function. Can take either "rel", "common", "random", or be assigned a numeric value (see details).
ed50	Refers to the ED50 parameter of the Emax dose-response function. Can take either "rel", "common", "random", or be assigned a numeric value (see details).
method	Can take either "common" or "random" to indicate whether relative effects should be modelled with between-study heterogeneity or not (see details).
class.effect	A list of named strings that determines which dose-response parameters to model with a class effect and what that effect should be ("common" or "random"). Element names should match dose-response parameter names (which will therefore depend on whether or not a wrapper function has been used for mbnma.run()). For example: list("beta.2"="fixed","beta.3"="random") when using mbnma.run() or list("ed50"="fixed","hill"="random") when using mbnma.emax.hill().
UME	A boolean object to indicate whether to fit an Unrelated Mean Effects model that does not assume consistency and so can be used to test if the consistency assumption is valid.
cor	A boolean object that indicates whether correlation should be modelled between relative effect dose-response parameters (TRUE) or not (FALSE). This is automatically set to FALSE if class effects are modelled or if multiple dose-response functions are fitted.
var.scale	A numeric vector indicating the relative scale of variances between correlated dose-response parameters when relative effects are modelled on more than one dose-response parameter and cor=TRUE (see details). Each element of the vector refers to the relative scale of each of the dose-response parameters that is modelled using relative effects.
parameters.to.save	A character vector containing names of parameters to monitor in JAGS
pd	Can take either: • pv only pV will be reported (as automatically outputted by R2jags). • plugin calculates pD by the plug-in method (Spiegelhalter et al. 2002). It is faster, but may output negative non-sensical values, due to skewed deviances that can arise with non-linear models. • pd.kl calculates pD by the Kullback-Leibler divergence (Plummer 2008). This will require running the model for additional iterations but will always produce a positive result. • popt calculates pD using an optimism adjustment which allows for calculation of the penalized expected deviance (Plummer 2008)
parallel	A boolean value that indicates whether JAGS should be run in parallel (TRUE) or not (FALSE). If TRUE then the number of cores to use is automatically calculated.

likelihood	A string indicating the likelihood to use in the model. Can take either "binomial", "normal" or "poisson". If left as NULL the likelihood will be inferred from the data.
link	A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. "logit", "log", "probit", "cloglog") or be assigned the value "identity" for and identity link function. If left as NULL the link function will be automatically assigned based on the likelihood.
priors	A named list of parameter values (without indices) and replacement prior distribution values given as strings **using distributions as specified in JAGS syntax** (see examples).
arg.params	Contains a list of arguments sent to mbnma.run() by dose-response specific wrapper functions
...	Arguments to be sent to R2jags.

Value

An object of S3 class(c("mbnma","rjags")) containing parameter results from the model. Can be summarized by print() and can check traceplots using R2jags::traceplot() or various functions from the package mcmcplots.

Nodes that are automatically monitored (if present in the model) have the following interpretation. These will have an additional suffix that relates to the name/number of the dose-response parameter to which they correspond (e.g. d.ed50 or d.1):

- d The pooled effect for each agent for a given dose-response parameter. Will be estimated by the model if dose-response parameters (beta.1, beta.2, beta.3, beta.4) are set to "rel".
- sd (without a suffix) - the between-study SD (heterogeneity) for relative effects, reported if method="random".
- D The class effect for each class for a given dose-response parameter. Will be estimated by the model if specified in class.effect.
- sd.D The within-class SD for different agents within the same class. Will be estimated by the model if any dose-response parameter in class.effect is set to "random".
- beta The absolute value of a given dose-response parameter across the whole network (does not vary by agent/class). Will be estimated by the model if dose-response parameters (beta.1, beta.2, beta.3, beta.4) are set to "common" or "random".
- sd (with a suffix) - the between-study SD (heterogeneity) for absolute dose-response parameters, reported if beta.1, beta.2, beta.3 or beta.4 are set to "random"
- totresdev The residual deviance of the model
- deviance The deviance of the model

If there are errors in the JAGS model code then the object will be a list consisting of two elements - an error message from JAGS that can help with debugging and model.arg, a list of arguments provided to mbnma.run() which includes jagscode, the JAGS code for the model that can help users identify the source of the error.

Dose-response parameters

- `"rel"` implies that relative effects should be pooled for this dose-response parameter separately for each agent in the network.
- `"common"` implies that all studies estimate the same true absolute effect (akin to a "fixed effects" meta-analysis) across the whole network
- `"random"` implies that all studies estimate a separate true absolute effect, but that each of these true effects vary randomly around a true mean effect. This approach allows for modelling of between-study heterogeneity.
- `numeric()` Assigned a numeric value. It indicates that this dose-response parameter should not be estimated from the data but should be assigned the numeric value determined by the user. This can be useful for fixing specific dose-response parameters (e.g. Hill parameters in Emax functions) to a value.

References

Mawdsley D, Bennetts M, Dias S, Boucher M, Welton NJ (2016). "Model-Based Network Meta-Analysis: A Framework for Evidence Synthesis of Clinical Trial Data." *CPT Pharmacometrics Syst Pharmacol*, **5**(8), 393-401. ISSN 2163-8306 (Electronic) 2163-8306 (Linking), doi: 10.1002/psp4.12091, https://pubmed.ncbi.nlm.nih.gov/27479782/.

Plummer M (2008). "Penalized loss functions for Bayesian model comparison." *Biostatistics*, **9**(3), 523-39. ISSN 1468-4357 (Electronic) 1465-4644 (Linking), https://pubmed.ncbi.nlm.nih.gov/18209015/.

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A (2002). "Bayesian measures of model complexity and fit." *J R Statistic Soc B*, **64**(4), 583-639.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Fit an Emax dose-response MBNMA with random treatment effects on Emax and ED50
emax <- mbnma.emax(network, emax="rel", ed50="rel", method="random")

# Fit an Emax dose-response MBNMA with common treatment effects on Emax and
#a single common parameter estimated for ED50
emax <- mbnma.emax(network, emax="rel", ed50="common", method="common")


########## Class effects ##########

# Generate a dataset with one class for active treatments and one for placebo
class.df <- HF2PPITT
class.df$class <- ifelse(class.df$agent=="placebo", "placebo", "active")
netclass <- mbnma.network(class.df)

# Fit an Emax function with common relative effects on Emax and ED50 and
```

```
#a random class effect on ED50.
emax <- mbnma.emax(netclass,
            emax="rel", ed50="rel", method="common",
            class.effect=list(ed50="random"))


####### Priors #######

# Obtain priors from an Emax function with random relative effects on Emax and ED50
emax <- mbnma.emax(network,
            emax="rel", ed50="rel", method="random")
print(emax$model.arg$priors)

# Set new more informative prior distributions
newpriors <- list(sd = "dnorm(0,0.5) T(0,)",
                  inv.R = "dwish(Omega[,],100)")

emax <- mbnma.emax(network,
            emax="rel", ed50="rel", method="random",
            priors=newpriors)


########## Sampler options ##########

# Change the number of MCMC iterations, the number of chains, and the thin
emax <- mbnma.emax(network, emax="rel", ed50="rel",
            n.iter=5000, n.thin=5, n.chains=4)

# Calculate effective number of parameters via plugin method
emax <- mbnma.emax(network, emax="rel", ed50="rel",
            pd="plugin")

# Calculate effective number of parameters via Kullback-Leibler method
emax <- mbnma.emax(network, emax="rel", ed50="rel",
            pd="pd.kl")


####### Examine MCMC diagnostics (using mcmcplots package) #######

# Density plots
mcmcplots::denplot(emax)

# Traceplots
mcmcplots::traplot(emax)

# Caterpillar plots
mcmcplots::caterplot(emax, "d.emax")


########## Output ###########

# Print R2jags output and summary
print(emax)
```

```
summary(emax)

# Plot forest plot of results
plot(emax)
```

`mbnma.emax.hill` *Run MBNMA model with an Emax dose-response function (with a Hill parameter)*

Description

Fits a Bayesian model-based network meta-analysis (MBNMA) with a defined dose-response function. Follows the methods of Mawdsley et al. (2016). This function acts as a wrapper for `mbnma.run()` that uses more clearly defined parameter names.

Usage

```
mbnma.emax.hill(
  network,
  emax = "rel",
  ed50 = "rel",
  hill = "common",
  method = "common",
  class.effect = list(),
  UME = FALSE,
  cor = TRUE,
  var.scale = NULL,
  parameters.to.save = NULL,
  pd = "pv",
  parallel = FALSE,
  likelihood = NULL,
  link = NULL,
  priors = NULL,
  arg.params = NULL,
  ...
)
```

Arguments

`network`	An object of class `mbnma.network`.
`emax`	Refers to the Emax parameter of the Emax dose-response function. Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).
`ed50`	Refers to the ED50 parameter of the Emax dose-response function. Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).

`hill` Refers to the Hill parameter of the Emax dose-response function. Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).

`method` Can take either `"common"` or `"random"` to indicate whether relative effects should be modelled with between-study heterogeneity or not (see details).

`class.effect` A list of named strings that determines which dose-response parameters to model with a class effect and what that effect should be (`"common"` or `"random"`). Element names should match dose-response parameter names (which will therefore depend on whether or not a wrapper function has been used for `mbnma.run()`). For example: `list("beta.2"="fixed","beta.3"="random")` when using `mbnma.run()` or `list("ed50"="fixed","hill"="random")` when using `mbnma.emax.hill()`.

`UME` A boolean object to indicate whether to fit an Unrelated Mean Effects model that does not assume consistency and so can be used to test if the consistency assumption is valid.

`cor` A boolean object that indicates whether correlation should be modelled between relative effect dose-response parameters (`TRUE`) or not (`FALSE`). This is automatically set to `FALSE` if class effects are modelled or if multiple dose-response functions are fitted.

`var.scale` A numeric vector indicating the relative scale of variances between correlated dose-response parameters when relative effects are modelled on more than one dose-response parameter and `cor=TRUE` (see details). Each element of the vector refers to the relative scale of each of the dose-response parameters that is modelled using relative effects.

`parameters.to.save` A character vector containing names of parameters to monitor in JAGS

`pd` Can take either:

- `pv` only pV will be reported (as automatically outputted by `R2jags`).
- `plugin` calculates pD by the plug-in method (Spiegelhalter et al. 2002). It is faster, but may output negative non-sensical values, due to skewed deviances that can arise with non-linear models.
- `pd.kl` calculates pD by the Kullback-Leibler divergence (Plummer 2008). This will require running the model for additional iterations but will always produce a positive result.
- `popt` calculates pD using an optimism adjustment which allows for calculation of the penalized expected deviance (Plummer 2008)

`parallel` A boolean value that indicates whether JAGS should be run in parallel (`TRUE`) or not (`FALSE`). If `TRUE` then the number of cores to use is automatically calculated.

`likelihood` A string indicating the likelihood to use in the model. Can take either `"binomial"`, `"normal"` or `"poisson"`. If left as `NULL` the likelihood will be inferred from the data.

`link` A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. `"logit"`, `"log"`, `"probit"`, `"cloglog"`) or be assigned the value `"identity"` for and identity link function. If left as `NULL` the link function will be automatically assigned based on the likelihood.

`priors` A named list of parameter values (without indices) and replacement prior distribution values given as strings **using distributions as specified in JAGS syntax** (see examples).

arg.params	Contains a list of arguments sent to mbnma.run() by dose-response specific wrapper functions
...	Arguments to be sent to R2jags.

Value

An object of S3 class(c("mbnma","rjags")) containing parameter results from the model. Can be summarized by print() and can check traceplots using R2jags::traceplot() or various functions from the package mcmcplots.

Nodes that are automatically monitored (if present in the model) have the following interpretation. These will have an additional suffix that relates to the name/number of the dose-response parameter to which they correspond (e.g. d.ed50 or d.1):

- d The pooled effect for each agent for a given dose-response parameter. Will be estimated by the model if dose-response parameters (beta.1, beta.2, beta.3, beta.4) are set to "rel".
- sd (without a suffix) - the between-study SD (heterogeneity) for relative effects, reported if method="random".
- D The class effect for each class for a given dose-response parameter. Will be estimated by the model if specified in class.effect.
- sd.D The within-class SD for different agents within the same class. Will be estimated by the model if any dose-response parameter in class.effect is set to "random".
- beta The absolute value of a given dose-response parameter across the whole network (does not vary by agent/class). Will be estimated by the model if dose-response parameters (beta.1, beta.2, beta.3, beta.4) are set to "common" or "random".
- sd (with a suffix) - the between-study SD (heterogeneity) for absolute dose-response parameters, reported if beta.1, beta.2, beta.3 or beta.4 are set to "random"
- totresdev The residual deviance of the model
- deviance The deviance of the model

If there are errors in the JAGS model code then the object will be a list consisting of two elements - an error message from JAGS that can help with debugging and model.arg, a list of arguments provided to mbnma.run() which includes jagscode, the JAGS code for the model that can help users identify the source of the error.

Dose-response parameters

- "rel" implies that relative effects should be pooled for this dose-response parameter separately for each agent in the network.
- "common" implies that all studies estimate the same true absolute effect (akin to a "fixed effects" meta-analysis) across the whole network
- "random" implies that all studies estimate a separate true absolute effect, but that each of these true effects vary randomly around a true mean effect. This approach allows for modelling of between-study heterogeneity.
- numeric() Assigned a numeric value. It indicates that this dose-response parameter should not be estimated from the data but should be assigned the numeric value determined by the user. This can be useful for fixing specific dose-response parameters (e.g. Hill parameters in Emax functions) to a value.

References

Mawdsley D, Bennetts M, Dias S, Boucher M, Welton NJ (2016). "Model-Based Network Meta-Analysis: A Framework for Evidence Synthesis of Clinical Trial Data." *CPT Pharmacometrics Syst Pharmacol*, **5**(8), 393-401. ISSN 2163-8306 (Electronic) 2163-8306 (Linking), doi: 10.1002/psp4.12091, https://pubmed.ncbi.nlm.nih.gov/27479782/.

Plummer M (2008). "Penalized loss functions for Bayesian model comparison." *Biostatistics*, **9**(3), 523-39. ISSN 1468-4357 (Electronic) 1465-4644 (Linking), https://pubmed.ncbi.nlm.nih.gov/18209015/.

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A (2002). "Bayesian measures of model complexity and fit." *J R Statistic Soc B*, **64**(4), 583-639.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Fit an Emax (with Hill parameter) dose-response MBNMA with random treatment
#effects on Emax, ED50 and Hill
emax.hill <- mbnma.emax.hill(network, emax="rel", ed50="rel", hill="rel",
                 method="random")

# Fit an Emax (with Hill parameter) dose-response MBNMA with common treatment
#effects on Emax, a single random parameter estimated for ED50
#and a single common parameter estimated for Hill
emax.hill <- mbnma.emax.hill(network, emax="rel", ed50="random", hill="common",
                 method="common")

# Assign a specific numerical value for Hill parameter
emax.hill <- mbnma.emax.hill(network, emax="rel", ed50="rel", hill=5)


########## Class effects ##########

# Generate a dataset with one class for active treatments and one for placebo
class.df <- HF2PPITT
class.df$class <- ifelse(class.df$agent=="placebo", "placebo", "active")
netclass <- mbnma.network(class.df)

# Fit an Emax (with Hill parameter) function with common relative effects on
#all parameters and common class effects on ED50 and Hill.
emax.hill <- mbnma.emax.hill(netclass,
                 emax="rel", ed50="rel", hill="rel", method="common",
                 class.effect=list(ed50="common", hill="common"))


####### Priors #######

# Obtain priors from an Emax (with Hill parameter) function with
```

```
#relative effects on Emax and ED50 and a single common parameter for Hill
emax.hill <- mbnma.emax.hill(network,
                 emax="rel", ed50="rel", hill="common", method="common")
print(emax.hill$model.arg$priors)

# Set new more informative prior distributions
newpriors <- list(beta.hill = "dnorm(0,0.5) T(,0)")

emax.hill <- mbnma.emax.hill(network,
                 emax="rel", ed50="rel", hill="common", method="common",
                 priors=newpriors)


########## Sampler options ##########

# Change the number of MCMC iterations, the number of chains, and the thin
emax.hill <- mbnma.emax.hill(network, emax="rel", ed50="rel", hill=2,
                 n.iter=5000, n.thin=5, n.chains=4)

# Calculate effective number of parameters via plugin method
emax.hill <- mbnma.emax.hill(network, emax="rel", ed50="rel", hill=2,
                 pd="plugin")

# Calculate effective number of parameters via Kullback-Leibler method
emax.hill <- mbnma.emax.hill(network, emax="rel", ed50="rel", hill=2,
                 pd="pd.kl")


####### Examine MCMC diagnostics (using mcmcplots package) #######

# Density plots
mcmcplots::denplot(emax.hill)

# Traceplots
mcmcplots::traplot(emax.hill)

# Caterpillar plots
mcmcplots::caterplot(emax.hill, "d.emax")


########## Output ###########

# Print R2jags output and summary
print(emax.hill)
summary(emax.hill)

# Plot forest plot of results
plot(emax.hill)
```

`mbnma.exponential` *Run MBNMA model with a exponential dose-response function*

Description

Fits a Bayesian model-based network meta-analysis (MBNMA) with a defined dose-response function. Follows the methods of Mawdsley et al. (2016). This function acts as a wrapper for `mbnma.run()` that uses more clearly defined parameter names.

Usage

```
mbnma.exponential(
  network,
  lambda = "rel",
  method = "common",
  class.effect = list(),
  UME = FALSE,
  cor = TRUE,
  var.scale = NULL,
  parameters.to.save = NULL,
  pd = "pv",
  parallel = FALSE,
  likelihood = NULL,
  link = NULL,
  priors = NULL,
  arg.params = NULL,
  ...
)
```

Arguments

`network`	An object of class `mbnma.network`.
`lambda`	Refers to the rate of growth/decay of the exponential dose-response function. Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).
`method`	Can take either `"common"` or `"random"` to indicate whether relative effects should be modelled with between-study heterogeneity or not (see details).
`class.effect`	A list of named strings that determines which dose-response parameters to model with a class effect and what that effect should be (`"common"` or `"random"`). Element names should match dose-response parameter names (which will therefore depend on whether or not a wrapper function has been used for `mbnma.run()`). For example: `list("beta.2"="fixed","beta.3"="random")` when using `mbnma.run()` or `list("ed50"="fixed","hill"="random")` when using `mbnma.emax.hill()`.
`UME`	A boolean object to indicate whether to fit an Unrelated Mean Effects model that does not assume consistency and so can be used to test if the consistency assumption is valid.

`cor` A boolean object that indicates whether correlation should be modelled between relative effect dose-response parameters (`TRUE`) or not (`FALSE`). This is automatically set to `FALSE` if class effects are modelled or if multiple dose-response functions are fitted.

`var.scale` A numeric vector indicating the relative scale of variances between correlated dose-response parameters when relative effects are modelled on more than one dose-response parameter and `cor=TRUE` (see details). Each element of the vector refers to the relative scale of each of the dose-response parameters that is modelled using relative effects.

`parameters.to.save`
A character vector containing names of parameters to monitor in JAGS

`pd` Can take either:

- `pv` only pV will be reported (as automatically outputted by `R2jags`).
- `plugin` calculates pD by the plug-in method (Spiegelhalter et al. 2002). It is faster, but may output negative non-sensical values, due to skewed deviances that can arise with non-linear models.
- `pd.kl` calculates pD by the Kullback-Leibler divergence (Plummer 2008). This will require running the model for additional iterations but will always produce a positive result.
- `popt` calculates pD using an optimism adjustment which allows for calculation of the penalized expected deviance (Plummer 2008)

`parallel` A boolean value that indicates whether JAGS should be run in parallel (`TRUE`) or not (`FALSE`). If `TRUE` then the number of cores to use is automatically calculated.

`likelihood` A string indicating the likelihood to use in the model. Can take either `"binomial"`, `"normal"` or `"poisson"`. If left as `NULL` the likelihood will be inferred from the data.

`link` A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. `"logit"`, `"log"`, `"probit"`, `"cloglog"`) or be assigned the value `"identity"` for and identity link function. If left as `NULL` the link function will be automatically assigned based on the likelihood.

`priors` A named list of parameter values (without indices) and replacement prior distribution values given as strings **using distributions as specified in JAGS syntax** (see examples).

`arg.params` Contains a list of arguments sent to `mbnma.run()` by dose-response specific wrapper functions

`...` Arguments to be sent to R2jags.

Value

An object of S3 `class(c("mbnma","rjags"))` containing parameter results from the model. Can be summarized by `print()` and can check traceplots using `R2jags::traceplot()` or various functions from the package `mcmcplots`.

Nodes that are automatically monitored (if present in the model) have the following interpretation. These will have an additional suffix that relates to the name/number of the dose-response parameter to which they correspond (e.g. `d.ed50` or `d.1`):

- d The pooled effect for each agent for a given dose-response parameter. Will be estimated by the model if dose-response parameters (beta.1, beta.2, beta.3, beta.4) are set to "rel".
- sd (without a suffix) - the between-study SD (heterogeneity) for relative effects, reported if method="random".
- D The class effect for each class for a given dose-response parameter. Will be estimated by the model if specified in class.effect.
- sd.D The within-class SD for different agents within the same class. Will be estimated by the model if any dose-response parameter in class.effect is set to "random".
- beta The absolute value of a given dose-response parameter across the whole network (does not vary by agent/class). Will be estimated by the model if dose-response parameters (beta.1, beta.2, beta.3, beta.4) are set to "common" or "random".
- sd (with a suffix) - the between-study SD (heterogeneity) for absolute dose-response parameters, reported if beta.1, beta.2, beta.3 or beta.4 are set to "random"
- totresdev The residual deviance of the model
- deviance The deviance of the model

If there are errors in the JAGS model code then the object will be a list consisting of two elements - an error message from JAGS that can help with debugging and model.arg, a list of arguments provided to mbnma.run() which includes jagscode, the JAGS code for the model that can help users identify the source of the error.

Dose-response parameters

- "rel" implies that relative effects should be pooled for this dose-response parameter separately for each agent in the network.
- "common" implies that all studies estimate the same true absolute effect (akin to a "fixed effects" meta-analysis) across the whole network
- "random" implies that all studies estimate a separate true absolute effect, but that each of these true effects vary randomly around a true mean effect. This approach allows for modelling of between-study heterogeneity.
- numeric() Assigned a numeric value. It indicates that this dose-response parameter should not be estimated from the data but should be assigned the numeric value determined by the user. This can be useful for fixing specific dose-response parameters (e.g. Hill parameters in Emax functions) to a value.

References

Mawdsley D, Bennetts M, Dias S, Boucher M, Welton NJ (2016). "Model-Based Network Meta-Analysis: A Framework for Evidence Synthesis of Clinical Trial Data." *CPT Pharmacometrics Syst Pharmacol*, **5**(8), 393-401. ISSN 2163-8306 (Electronic) 2163-8306 (Linking), doi: 10.1002/psp4.12091, https://pubmed.ncbi.nlm.nih.gov/27479782/.

Plummer M (2008). "Penalized loss functions for Bayesian model comparison." *Biostatistics*, **9**(3), 523-39. ISSN 1468-4357 (Electronic) 1465-4644 (Linking), https://pubmed.ncbi.nlm.nih.gov/18209015/.

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A (2002). "Bayesian measures of model complexity and fit." *J R Statistic Soc B*, **64**(4), 583-639.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Fit a exponential dose-response MBNMA with random treatment effects
exponential <- mbnma.exponential(network, lambda="rel", method="random")

# Fit a exponential dose-response MBNMA using a cloglog link function
exponential <- mbnma.exponential(network, lambda="rel", link="cloglog")


####### Priors #######

# Obtain priors from exponential dose-response MBNMA
exponential <- mbnma.exponential(network, lambda="rel", method="random")
print(exponential$model.arg$priors)

# Set new more informative prior distributions
newpriors <- list(sd = "dnorm(0,0.5) T(0,)")

exponential <- mbnma.exponential(network, lambda="rel", method="random",
                  priors=newpriors)


########## Sampler options ##########

# Change the number of MCMC iterations, the number of chains, and the thin
exponential <- mbnma.exponential(network, lambda="rel", method="random",
                  n.iter=5000, n.thin=5, n.chains=4)

# Calculate effective number of parameters via plugin method
exponential <- mbnma.exponential(network, lambda="rel", method="random",
                  pd="plugin")

# Calculate effective number of parameters via Kullback-Leibler method
exponential <- mbnma.exponential(network, lambda="rel", method="random",
                  pd="pd.kl")


####### Examine MCMC diagnostics (using mcmcplots package) #######

# Density plots
mcmcplots::denplot(exponential)

# Traceplots
mcmcplots::traplot(exponential)

# Caterpillar plots
mcmcplots::caterplot(exponential, "d.lambda")
```

```
########## Output ###########

# Print R2jags output and summary
print(exponential)
summary(exponential)

# Plot forest plot of results
plot(exponential)
```

`mbnma.linear` *Run MBNMA model with a linear dose-response function*

Description

Fits a Bayesian model-based network meta-analysis (MBNMA) with a defined dose-response function. Follows the methods of Mawdsley et al. (2016). This function acts as a wrapper for `mbnma.run()` that uses more clearly defined parameter names.

Usage

```
mbnma.linear(
  network,
  slope = "rel",
  method = "common",
  class.effect = list(),
  UME = FALSE,
  cor = TRUE,
  var.scale = NULL,
  parameters.to.save = NULL,
  pd = "pv",
  parallel = FALSE,
  likelihood = NULL,
  link = NULL,
  priors = NULL,
  arg.params = NULL,
  ...
)
```

Arguments

`network`	An object of class `mbnma.network`.
`slope`	Refers to the slope parameter of the linear dose-response function. Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).
`method`	Can take either `"common"` or `"random"` to indicate whether relative effects should be modelled with between-study heterogeneity or not (see details).

`class.effect` A list of named strings that determines which dose-response parameters to model with a class effect and what that effect should be (`"common"` or `"random"`). Element names should match dose-response parameter names (which will therefore depend on whether or not a wrapper function has been used for `mbnma.run()`). For example: `list("beta.2"="fixed","beta.3"="random")` when using `mbnma.run()` or `list("ed50"="fixed","hill"="random")` when using `mbnma.emax.hill()`.

`UME` A boolean object to indicate whether to fit an Unrelated Mean Effects model that does not assume consistency and so can be used to test if the consistency assumption is valid.

`cor` A boolean object that indicates whether correlation should be modelled between relative effect dose-response parameters (`TRUE`) or not (`FALSE`). This is automatically set to `FALSE` if class effects are modelled or if multiple dose-response functions are fitted.

`var.scale` A numeric vector indicating the relative scale of variances between correlated dose-response parameters when relative effects are modelled on more than one dose-response parameter and `cor=TRUE` (see details). Each element of the vector refers to the relative scale of each of the dose-response parameters that is modelled using relative effects.

`parameters.to.save` A character vector containing names of parameters to monitor in JAGS

`pd` Can take either:

- `pv` only pV will be reported (as automatically outputted by `R2jags`).
- `plugin` calculates pD by the plug-in method (Spiegelhalter et al. 2002). It is faster, but may output negative non-sensical values, due to skewed deviances that can arise with non-linear models.
- `pd.kl` calculates pD by the Kullback-Leibler divergence (Plummer 2008). This will require running the model for additional iterations but will always produce a positive result.
- `popt` calculates pD using an optimism adjustment which allows for calculation of the penalized expected deviance (Plummer 2008)

`parallel` A boolean value that indicates whether JAGS should be run in parallel (`TRUE`) or not (`FALSE`). If `TRUE` then the number of cores to use is automatically calculated.

`likelihood` A string indicating the likelihood to use in the model. Can take either `"binomial"`, `"normal"` or `"poisson"`. If left as `NULL` the likelihood will be inferred from the data.

`link` A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. `"logit"`, `"log"`, `"probit"`, `"cloglog"`) or be assigned the value `"identity"` for and identity link function. If left as `NULL` the link function will be automatically assigned based on the likelihood.

`priors` A named list of parameter values (without indices) and replacement prior distribution values given as strings **using distributions as specified in JAGS syntax** (see examples).

`arg.params` Contains a list of arguments sent to `mbnma.run()` by dose-response specific wrapper functions

`...` Arguments to be sent to R2jags.

Value

An object of S3 `class(c("mbnma","rjags"))` containing parameter results from the model. Can be summarized by `print()` and can check traceplots using `R2jags::traceplot()` or various functions from the package `mcmcplots`.

Nodes that are automatically monitored (if present in the model) have the following interpretation. These will have an additional suffix that relates to the name/number of the dose-response parameter to which they correspond (e.g. `d.ed50` or `d.1`):

- `d` The pooled effect for each agent for a given dose-response parameter. Will be estimated by the model if dose-response parameters (`beta.1`, `beta.2`, `beta.3`, `beta.4`) are set to `"rel"`.
- `sd` (without a suffix) - the between-study SD (heterogeneity) for relative effects, reported if `method="random"`.
- `D` The class effect for each class for a given dose-response parameter. Will be estimated by the model if specified in `class.effect`.
- `sd.D` The within-class SD for different agents within the same class. Will be estimated by the model if any dose-response parameter in `class.effect` is set to `"random"`.
- `beta` The absolute value of a given dose-response parameter across the whole network (does not vary by agent/class). Will be estimated by the model if dose-response parameters (`beta.1`, `beta.2`, `beta.3`, `beta.4`) are set to `"common"` or `"random"`.
- `sd` (with a suffix) - the between-study SD (heterogeneity) for absolute dose-response parameters, reported if `beta.1`, `beta.2`, `beta.3` or `beta.4` are set to `"random"`
- `totresdev` The residual deviance of the model
- `deviance` The deviance of the model

If there are errors in the JAGS model code then the object will be a list consisting of two elements - an error message from JAGS that can help with debugging and `model.arg`, a list of arguments provided to `mbnma.run()` which includes `jagscode`, the JAGS code for the model that can help users identify the source of the error.

Dose-response parameters

- `"rel"` implies that relative effects should be pooled for this dose-response parameter separately for each agent in the network.
- `"common"` implies that all studies estimate the same true absolute effect (akin to a "fixed effects" meta-analysis) across the whole network
- `"random"` implies that all studies estimate a separate true absolute effect, but that each of these true effects vary randomly around a true mean effect. This approach allows for modelling of between-study heterogeneity.
- `numeric()` Assigned a numeric value. It indicates that this dose-response parameter should not be estimated from the data but should be assigned the numeric value determined by the user. This can be useful for fixing specific dose-response parameters (e.g. Hill parameters in Emax functions) to a value.

References

Mawdsley D, Bennetts M, Dias S, Boucher M, Welton NJ (2016). "Model-Based Network Meta-Analysis: A Framework for Evidence Synthesis of Clinical Trial Data." *CPT Pharmacometrics Syst Pharmacol*, **5**(8), 393-401. ISSN 2163-8306 (Electronic) 2163-8306 (Linking), doi: 10.1002/psp4.12091, https://pubmed.ncbi.nlm.nih.gov/27479782/.

Plummer M (2008). "Penalized loss functions for Bayesian model comparison." *Biostatistics*, **9**(3), 523-39. ISSN 1468-4357 (Electronic) 1465-4644 (Linking), https://pubmed.ncbi.nlm.nih.gov/18209015/.

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A (2002). "Bayesian measures of model complexity and fit." *J R Statistic Soc B*, **64**(4), 583-639.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Fit a linear dose-response MBNMA with random treatment effects
linear <- mbnma.linear(network, slope="rel", method="random")

# Fit a linear dose-response MBNMA using a cloglog link function
linear <- mbnma.linear(network, slope="rel", link="cloglog")


####### Priors #######

# Obtain priors from linear dose-response MBNMA
linear <- mbnma.linear(network, slope="rel", method="random")
print(linear$model.arg$priors)

# Set new more informative prior distributions
newpriors <- list(sd = "dnorm(0,0.5) T(0,)")

linear <- mbnma.linear(network, slope="rel", method="random",
              priors=newpriors)


########## Sampler options ##########

# Change the number of MCMC iterations, the number of chains, and the thin
linear <- mbnma.linear(network, slope="rel", method="random",
              n.iter=5000, n.thin=5, n.chains=4)

# Calculate effective number of parameters via plugin method
linear <- mbnma.linear(network, slope="rel", method="random",
              pd="plugin")

# Calculate effective number of parameters via Kullback-Leibler method
linear <- mbnma.linear(network, slope="rel", method="random",
```

```
              pd="pd.kl")


####### Examine MCMC diagnostics (using mcmcplots package) #######

# Density plots
mcmcplots::denplot(linear)

# Traceplots
mcmcplots::traplot(linear)

# Caterpillar plots
mcmcplots::caterplot(linear, "d.slope")


########## Output ###########

# Print R2jags output and summary
print(linear)
summary(linear)

# Plot forest plot of results
plot(linear)
```

`mbnma.nodesplit` *Node-splitting model for testing consistency at the treatment level using MBNMA*

Description

Splits contributions for a given set of treatment comparisons into direct and indirect evidence. A discrepancy between the two suggests that the consistency assumption required for NMA and MBNMA may violated.

Usage

```
mbnma.nodesplit(
  network,
  fun = "linear",
  user.fun = NULL,
  beta.1 = "rel",
  beta.2 = "rel",
  beta.3 = "rel",
  beta.4 = "rel",
  method = "common",
  knots = 3,
  comparisons = NULL,
```

```
  incldr = TRUE,
  ...
)

## S3 method for class 'nodesplit'
plot(x, plot.type = NULL, ...)
```

Arguments

network	An object of class mbnma.network.
fun	A character vector specifying a functional form to be assigned to the dose-response. Options are given in details.
user.fun	A formula specifying any relationship including dose and one/several of: beta.1, beta.2, beta.3, beta.4.
beta.1	Refers to dose-parameter(s) specified within the dose-response function(s). Can take either "rel", "common", "random", or be assigned a numeric value (see details).
beta.2	Refers to dose-parameter(s) specified within the dose-response function(s). Can take either "rel", "common", "random", or be assigned a numeric value (see details).
beta.3	Refers to dose-parameter(s) specified within the dose-response function(s). Can take either "rel", "common", "random", or be assigned a numeric value (see details).
beta.4	Refers to dose-parameter(s) specified within the dose-response function(s). Can take either "rel", "common", "random", or be assigned a numeric value (see details).
method	Can take either "common" or "random" to indicate whether relative effects should be modelled with between-study heterogeneity or not (see details).
knots	The number/location of knots if a restricted cubic spline dose-response function is fitted (fun="rcs"). If a single number is given it indicates the number of knots (they will be equally spaced across the range of doses). If a numeric vector is given it indicates the location of the knots. Minimum number of knots is 3.
comparisons	A matrix specifying the comparisons to be split (one row per comparison). The matrix must have two columns indicating each treatment for each comparison. Values can either be character (corresponding to the treatment names given in network) or numeric (corresponding to treatment codes within the network - note that these may change if drop.discon = TRUE).
incldr	A boolean object indicating whether or not to allow for indirect evidence contributions via the dose-response relationship. This can be used when node-splitting in dose-response MBNMA to allow for a greater number of potential loops in which to check for consistency.
...	Arguments to be sent to ggplot2::ggplot()
x	An object of class("nodesplit")
plot.type	A character string that can take the value of "forest" to plot only forest plots, "density" to plot only density plots, or left as NULL (the default) to plot both types of plot.

Details

The S3 method `plot()` on an `nodesplit` object generates either forest plots of posterior medians and 95\% credible intervals, or density plots of posterior densities for direct and indirect evidence.

Value

Plots the desired graph(s) and returns an object (or list of object if `plot.type=NULL`) of `class(c("gg","ggplot"))`

Methods (by generic)

- `plot`: Plot outputs from treatment-level nodesplit MBNMA models

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

split <- mbnma.nodesplit(network, fun="emax", likelihood = "binomial", link="logit",
  method="common")


#### To perform nodesplit on selected comparisons ####

# Check for closed loops of treatments with independent evidence sources
# Including indirect evidence via the dose-response relationship
loops <- inconsistency.loops(network$data.ab, incldr=TRUE)

# This...
single.split <- mbnma.nodesplit(network, fun="exponential", likelihood = "binomial", link="logit",
             method="random", comparisons=rbind(c("sumatriptan_1", "almotriptan_1")))

#...is the same as...
single.split <- mbnma.nodesplit(network, fun="exponential", likelihood = "binomial", link="logit",
             method="random", comparisons=rbind(c(6, 12)))


# Plot results
plot(split, plot.type="density") # Plot density plots of posterior densities
plot(split, plot.type="forest") # Plot forest plots of direct and indirect evidence

# Print and summarise results
print(split)
summary(split) # Generate a data frame of summary results
```

`mbnma.run` *Run MBNMA dose-response models*

Description

Fits a Bayesian dose-response for model-based network meta-analysis (MBNMA) that can account for multiple doses of different agents by applying a desired dose-response function. Follows the methods of Mawdsley et al. (2016).

Usage

```
mbnma.run(
  network,
  fun = "linear",
  beta.1 = "rel",
  beta.2 = "rel",
  beta.3 = "rel",
  beta.4 = "rel",
  method = "common",
  class.effect = list(),
  UME = FALSE,
  knots = 3,
  cor = TRUE,
  var.scale = NULL,
  user.fun = NULL,
  parameters.to.save = NULL,
  pd = "pv",
  parallel = FALSE,
  likelihood = NULL,
  link = NULL,
  priors = NULL,
  model.file = NULL,
  n.iter = 10000,
  n.chains = 3,
  n.burnin = floor(n.iter/2),
  n.thin = max(1, floor((n.iter - n.burnin)/1000)),
  autojags = FALSE,
  Rhat = 1.1,
  n.update = 10,
  arg.params = NULL,
  ...
)
```

Arguments

`network`	An object of class `mbnma.network`.

`fun` A character vector specifying a functional form to be assigned to the dose-response. Options are given in `details`.

`beta.1` Refers to dose-parameter(s) specified within the dose-response function(s). Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).

`beta.2` Refers to dose-parameter(s) specified within the dose-response function(s). Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).

`beta.3` Refers to dose-parameter(s) specified within the dose-response function(s). Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).

`beta.4` Refers to dose-parameter(s) specified within the dose-response function(s). Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).

`method` Can take either `"common"` or `"random"` to indicate whether relative effects should be modelled with between-study heterogeneity or not (see details).

`class.effect` A list of named strings that determines which dose-response parameters to model with a class effect and what that effect should be (`"common"` or `"random"`). Element names should match dose-response parameter names (which will therefore depend on whether or not a wrapper function has been used for `mbnma.run()`). For example: `list("beta.2"="fixed","beta.3"="random")` when using `mbnma.run()` or `list("ed50"="fixed","hill"="random")` when using `mbnma.emax.hill()`.

`UME` A boolean object to indicate whether to fit an Unrelated Mean Effects model that does not assume consistency and so can be used to test if the consistency assumption is valid.

`knots` The number/location of knots if a restricted cubic spline dose-response function is fitted (`fun="rcs"`). If a single number is given it indicates the number of knots (they will be equally spaced across the range of doses). If a numeric vector is given it indicates the location of the knots. Minimum number of knots is 3.

`cor` A boolean object that indicates whether correlation should be modelled between relative effect dose-response parameters (`TRUE`) or not (`FALSE`). This is automatically set to `FALSE` if class effects are modelled or if multiple dose-response functions are fitted.

`var.scale` A numeric vector indicating the relative scale of variances between correlated dose-response parameters when relative effects are modelled on more than one dose-response parameter and `cor=TRUE` (see details). Each element of the vector refers to the relative scale of each of the dose-response parameters that is modelled using relative effects.

`user.fun` A formula specifying any relationship including `dose` and one/several of: `beta.1`, `beta.2`, `beta.3`, `beta.4`.

`parameters.to.save`
A character vector containing names of parameters to monitor in JAGS

`pd` Can take either:

- `pv` only pV will be reported (as automatically outputted by `R2jags`).

- `plugin` calculates pD by the plug-in method (Spiegelhalter et al. 2002). It is faster, but may output negative non-sensical values, due to skewed deviances that can arise with non-linear models.
- `pd.kl` calculates pD by the Kullback-Leibler divergence (Plummer 2008). This will require running the model for additional iterations but will always produce a positive result.
- `popt` calculates pD using an optimism adjustment which allows for calculation of the penalized expected deviance (Plummer 2008)

`parallel` A boolean value that indicates whether JAGS should be run in parallel (`TRUE`) or not (`FALSE`). If `TRUE` then the number of cores to use is automatically calculated.

`likelihood` A string indicating the likelihood to use in the model. Can take either `"binomial"`, `"normal"` or `"poisson"`. If left as `NULL` the likelihood will be inferred from the data.

`link` A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. `"logit"`, `"log"`, `"probit"`, `"cloglog"`) or be assigned the value `"identity"` for and identity link function. If left as `NULL` the link function will be automatically assigned based on the likelihood.

`priors` A named list of parameter values (without indices) and replacement prior distribution values given as strings **using distributions as specified in JAGS syntax** (see examples).

`model.file` A JAGS model written as a character object that can be used to overwrite the JAGS model that is automatically written based on the specified options.

`n.iter` number of total iterations per chain (including burn in; default: 15000)

`n.chains` number of Markov chains (default: 3)

`n.burnin` length of burn in, i.e. number of iterations to discard at the beginning. Default is ‘n.iter/2“, that is, discarding the first half of the simulations. If n.burnin is 0, jags() will run 100 iterations for adaption.

`n.thin` thinning rate. Must be a positive integer. Set n.thin > 1`` to save memory and computation time if n.iter is large. Default is max(1, floor(n.chains * (n.iter-n.burnin) / 1000))“ which will only thin if there are at least 2000 simulations.

`autojags` A boolean value that indicates whether the model should be continually updated until it has converged below a specific cutoff of `Rhat`

`Rhat` A cutoff value for the Gelman-Rubin convergence diagnostic(Gelman and Rubin 1992). Unless all parameters have Rhat values lower than this the model will continue to sequentially update up to a maximum of `n.update`. Default is `1.1`

`n.update` The maximum number of updates. Each update is run for 1000 iterations, after which the Rhat values of all parameters are checked against `Rhat`. Default maximum updates is `10` (i.e. 10,000 additional iterations in total).

`arg.params` Contains a list of arguments sent to `mbnma.run()` by dose-response specific wrapper functions

`...` Arguments to be sent to R2jags.

Details

When relative effects are modelled on more than one dose-response parameter and `cor = TRUE`, correlation between the dose-response parameters is automatically estimated using a vague Wishart prior. This prior can be made slightly more informative by specifying the relative scale of variances between the dose-response parameters using `var.scale`. `cor` will automatically be set to `FALSE` if class effects are modelled or if a model is fitted with multiple dose-response functions.

Value

An object of S3 `class(c("mbnma","rjags"))` containing parameter results from the model. Can be summarized by `print()` and can check traceplots using `R2jags::traceplot()` or various functions from the package `mcmcplots`.

Nodes that are automatically monitored (if present in the model) have the following interpretation. These will have an additional suffix that relates to the name/number of the dose-response parameter to which they correspond (e.g. `d.ed50` or `d.1`):

- `d` The pooled effect for each agent for a given dose-response parameter. Will be estimated by the model if dose-response parameters (`beta.1`, `beta.2`, `beta.3`, `beta.4`) are set to `"rel"`.
- `sd` (without a suffix) - the between-study SD (heterogeneity) for relative effects, reported if `method="random"`.
- `D` The class effect for each class for a given dose-response parameter. Will be estimated by the model if specified in `class.effect`.
- `sd.D` The within-class SD for different agents within the same class. Will be estimated by the model if any dose-response parameter in `class.effect` is set to `"random"`.
- `beta` The absolute value of a given dose-response parameter across the whole network (does not vary by agent/class). Will be estimated by the model if dose-response parameters (`beta.1`, `beta.2`, `beta.3`, `beta.4`) are set to `"common"` or `"random"`.
- `sd` (with a suffix) - the between-study SD (heterogeneity) for absolute dose-response parameters, reported if `beta.1`, `beta.2`, `beta.3` or `beta.4` are set to `"random"`
- `totresdev` The residual deviance of the model
- `deviance` The deviance of the model

If there are errors in the JAGS model code then the object will be a list consisting of two elements - an error message from JAGS that can help with debugging and `model.arg`, a list of arguments provided to `mbnma.run()` which includes `jagscode`, the JAGS code for the model that can help users identify the source of the error.

Dose-response parameters

- `"rel"` implies that relative effects should be pooled for this dose-response parameter separately for each agent in the network.
- `"common"` implies that all studies estimate the same true absolute effect (akin to a "fixed effects" meta-analysis) across the whole network
- `"random"` implies that all studies estimate a separate true absolute effect, but that each of these true effects vary randomly around a true mean effect. This approach allows for modelling of between-study heterogeneity.

- `numeric()` Assigned a numeric value. It indicates that this dose-response parameter should not be estimated from the data but should be assigned the numeric value determined by the user. This can be useful for fixing specific dose-response parameters (e.g. Hill parameters in Emax functions) to a value.

Dose-response function

Several general dose-response functions are provided, but a user-defined dose-response relationship can instead be used.

Built-in dose-response functions are:

- `"linear"`: `beta.1` refers to the gradient
- `"exponential"`: `beta.1` refers to the rate of gain/decay
- `"emax"` (emax without a Hill parameter): `beta.1` refers to Emax parameter, `beta.2` refers to ET50 parameter
- `"emax.hill"` (emax with a Hill parameter): `beta.1` refers to Emax parameter, `beta.2` refers to ET50 parameter, `beta.3` refers to Hill parameter
- `"rcs"` restricted cubic splines with knot number/location defined by `knot`.`beta.1` refers to the first spline coeffficient, `beta.2` to the second coefficient, etc. Follows the method of Hamza et al. (2020)
- `"nonparam.up"` (monotonically increasing non-parametric dose-response relationship following the method of Owen et al. (2015))
- `"nonparam.down"` (monotonically decreasing non-parametric dose-response relationship following the method of Owen et al. (2015))
- `"user"` (user-defined function: `user.fun` must be specified in arguments)

As of version 0.2.5, separate dose-response functions can be specified for different agents in the network by passing a character vector with multiple elements to `fun`. Each agent in `network` is assigned the dose-response function in the corresponding element in `fun`. `fun` must therefore be the same length as the number of agents in `network`. Dose-response parameters `beta.1`, `beta.2`, `beta.3` and `beta.4` refer to the corresponding dose-response parameters across the multiple functions in the following order: `user`, `linear`, `exponential`, `emax`, `emax.hill`.

This would mean that if `fun` included `linear`, `exponential` and `emax` within it then for the corresponding agents `beta.1` would refer to linear slope parameters, `beta.2` to exponential rate of growth/decay parameters, `beta.3` to Emax parameters, and `beta.4` to ED50 parameters.

References

Gelman A, Rubin DB (1992). "Inference from iterative simulation using multiple sequences." *Statistical Science*, **7**(4), 457-511. https://projecteuclid.org/euclid.ss/1177011136/.

Hamza T, Cipriani A, Furukawa TA, Egger M, Orsini N. Salanti G (2020). "A Bayesian dose-response meta-analysis model: simulation study and application." preprint, https://arxiv.org/abs/2004.12737v1.

Mawdsley D, Bennetts M, Dias S, Boucher M, Welton NJ (2016). "Model-Based Network Meta-Analysis: A Framework for Evidence Synthesis of Clinical Trial Data." *CPT Pharmacometrics*

Syst Pharmacol, **5**(8), 393-401. ISSN 2163-8306 (Electronic) 2163-8306 (Linking), doi: 10.1002/psp4.12091, https://pubmed.ncbi.nlm.nih.gov/27479782/.

Owen RK, Tincello DG, Keith RA (2015). "Network meta-analysis: development of a three-level hierarchical modeling approach incorporating dose-related constraints." *Value Health*, **18**(1), 116-26. ISSN 1524-4733 (Electronic) 1098-3015 (Linking), doi: 10.1016/j.jval.2014.10.006, https://pubmed.ncbi.nlm.nih.gov/25595242/.

Plummer M (2008). "Penalized loss functions for Bayesian model comparison." *Biostatistics*, **9**(3), 523-39. ISSN 1468-4357 (Electronic) 1465-4644 (Linking), https://pubmed.ncbi.nlm.nih.gov/18209015/.

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A (2002). "Bayesian measures of model complexity and fit." *J R Statistic Soc B*, **64**(4), 583-639.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)


######## Dose-response functions ########

# Fit a dose-response MBNMA with a linear function and common treatment effects
result <- mbnma.run(network, fun="linear", beta.1="rel", method="common")

# Fit a dose-response MBNMA with an exponential function and random treatment effects
result <- mbnma.run(network, fun="exponential", beta.1="rel", method="random")

# Fit a user-defined function (quadratic)
fun.def <- ~ (beta.1 * dose) + (beta.2 * (dose^2))
result <- mbnma.run(network, fun="user", user.fun=fun.def,
              beta.1="rel", beta.2="rel", method="common")

# Fit an Emax function with a single random (exchangeable) parameter estimated
#for ED50 and common treatment effects on relative Emax effects
result <- mbnma.run(network, fun="emax",
              beta.1="rel", beta.2="random", method="common")

# Fit an Emax function with a Hill parameter, with a fixed value for the Hill parameter
#provided to the model and random relative effects on Emax and ED50 (which will
#therefore be modelled with a correlation between them).
result <- mbnma.run(network, fun="emax.hill",
              beta.1="rel", beta.2="rel", beta.3=5, method="random")

# Fit a model with restricted cubic splines and 3 knots
#at 10% 30% and 60% quartiles of dose ranges
depnet <- mbnma.network(ssri) # Using the sSRI depression dataset
result <- mbnma.run(depnet, fun="rcs", knots=c(0.1,0.3,0.6))

# Fit a model with different dose-response functions for each agent
```

```
multidose <- mbnma.run(network, fun=c("emax", "emax", "emax", "exponential",
                "emax", "emax", "exponential", "emax"))


########## Class effects ##########

# Generate a dataset with one class for active treatments and one for placebo
class.df <- HF2PPITT
class.df$class <- ifelse(class.df$agent=="placebo", "placebo", "active")
netclass <- mbnma.network(class.df)

painnet <- mbnma.network(osteopain_2wkabs)

# Fit an Emax function with random relative effects on Emax and ED50 and
#a common class effect on Emax
result <- mbnma.run(painnet, fun="emax", method="random",
              beta.1="rel", beta.2="rel",
              class.effect=list(beta.1="common"))


####### Priors #######

# Obtain priors from an Emax function with random relative effects on Emax and ED50
result <- mbnma.run(network, fun="emax",
              beta.1="rel", beta.2="rel", method="random")
print(result$model.arg$priors)

# Set new more informative prior distributions
newpriors <- list(sd = "dnorm(0,0.5) T(0,)",
                  inv.R = "dwish(Omega[,],100)")

result <- mbnma.run(network, fun="emax",
              beta.1="rel", beta.2="rel", method="random",
              priors=newpriors)


########## Sampler options ##########

# Change the number of MCMC iterations, the number of chains, and the thin
result <- mbnma.run(network, fun="exponential", beta.1="rel", method="random",
              n.iter=5000, n.thin=5, n.chains=4)

# Calculate effective number of parameters via plugin method
result <- mbnma.run(network, fun="exponential", beta.1="rel", method="random",
              pd="plugin")

# Calculate effective number of parameters via Kullback-Leibler method
result <- mbnma.run(network, fun="exponential", beta.1="rel", method="random",
              pd="pd.kl")


####### Examine MCMC diagnostics (using mcmcplots package) #######
```

```
# Density plots
mcmcplots::denplot(result)

# Traceplots
mcmcplots::traplot(result)

# Caterpillar plots
mcmcplots::caterplot(result, "d.1")


####### Automatically run jags until convergence is reached #########

# Rhat of 1.08 is set as the criteria for convergence
#on all monitored parameters
conv.res <- mbnma.run(network, fun="emax",
              beta.1="rel", beta.2="rel", method="random",
              n.iter=10000, n.burnin=9000,
              autojags=TRUE, Rhat=1.08, n.update=8)


########## Output ############

# Print R2jags output and summary
print(result)
summary(result)

# Plot forest plot of results
plot(result)
```

`mbnma.update` *Update MBNMA to monitor deviance nodes in the model*

Description

Useful for obtaining deviance contributions or fitted values

Usage

```
mbnma.update(
  mbnma,
  param = "theta",
  n.iter = mbnma$BUGSoutput$n.iter,
  n.thin = mbnma$BUGSoutput$n.thin
)
```

Arguments

mbnma	An S3 object of class "mbnma" generated by running a dose-response MBNMA model
param	Used to indicate which node to monitor in the model. Can be any parameter in the model code that varies by all arms within all studies. These are some typical parameters that it might be of interest to monitor, provided they are in the original model code: • "theta" for fitted values • "psi" for fitted values on natural scale (e.g. probabilities) • "dev" for deviance contributions • "resdev" for residual deviance contributions • "delta" for within-study relative effects versus the study reference treatment
n.iter	number of total iterations per chain (including burn in; default: 2000)
n.thin	thinning rate. Must be a positive integer. Set n.thin > 1 to save memory and computation time if n.iter is large. Default is max(1,floor(n.chains * (n.iter-n.burnin) / 1000)) which will only thin if there are at least 2000 simulations.

Value

A data frame containing the posterior mean of the updates by arm and study, with arm and study identifiers.

For MBNMAdose:

- facet indicates the agent identifier in the given arm of a study
- fupdose indicates the dose in the given arm of a study

For MBNMAtime:

- facet indicates the treatment identifier in the given arm of the study
- fupdose indicates the follow-up time at the given observation in the given arm of the study

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Fit a dose-response MBNMA, monitoring "psi" and "resdev"
result <- mbnma.run(network, fun="exponential", beta.1="rel", method="random",
  parameters.to.save=c("psi", "resdev"))

mbnma.update(result, param="theta") # monitor theta

mbnma.update(result, param="rhat") # monitor rhat

mbnma.update(result, param="delta") # monitor delta
```

`mbnma.validate.data` *Validates that a dataset fulfils requirements for MBNMA*

Description

Validates that a dataset fulfils requirements for MBNMA

Usage

```
mbnma.validate.data(data.ab, single.arm = FALSE)
```

Arguments

`data.ab`	A data frame of arm-level data in "long" format containing the columns: • `studyID` Study identifiers • `dose` Numeric data indicating the dose (must take positive values) • `agent` Agent identifiers (can be numeric, factor or character) • `y` Numeric data indicating the aggregate response for a continuous outcome. Required for continuous data. • `se` Numeric data indicating the standard error for a given observation. Required for continuous data. • `r` Numeric data indicating the number of responders within a study arm. Required for binomial or poisson data. • `N` Numeric data indicating the total number of participants within a study arm. Required for binomial data • `E` Numeric data indicating the total exposure time for participants within a study arm. Required for poisson data. • `class` An optional column indicating a particular class code. Agents with the same identifier must also have the same class code.
`single.arm`	A boolean object to indicate whether to allow single arm studies in the dataset (`TRUE`) or not (`FALSE`)

Details

Checks done within the validation:

- Checks data.ab has required column names
- Checks there are no NAs
- Checks that all SEs are >0 (if variables are included in dataset)
- Checks that all doses are >=0
- Checks that all r and N are positive (if variables are included in dataset)

- Checks that all y, se, r, N and E are numeric
- Checks that class codes are consistent within each agent
- Checks that agent/class names do not contain restricted characters
- Checks that studies have at least two arms (if `single.arm = FALSE`)
- Checks that each study includes at least two treatments

Value

An error if checks are not passed. Runs silently if checks are passed

---

`mbnma.write` *Write MBNMA dose-response model JAGS code*

Description

Writes JAGS code for a Bayesian time-course model for model-based network meta-analysis (MBNMA).

Usage

```
mbnma.write(
  fun = "linear",
  beta.1 = "rel",
  beta.2 = NULL,
  beta.3 = NULL,
  beta.4 = NULL,
  method = "common",
  knots = 3,
  cor = TRUE,
  cor.prior = "wishart",
  var.scale = NULL,
  class.effect = list(),
  UME = FALSE,
  user.fun = NULL,
  likelihood = "binomial",
  link = NULL
)
```

Arguments

`fun`	A character vector specifying a functional form to be assigned to the dose-response. Options are given in `details`.
`beta.1`	Refers to dose-parameter(s) specified within the dose-response function(s). Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).

`beta.2` Refers to dose-parameter(s) specified within the dose-response function(s). Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).

`beta.3` Refers to dose-parameter(s) specified within the dose-response function(s). Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).

`beta.4` Refers to dose-parameter(s) specified within the dose-response function(s). Can take either `"rel"`, `"common"`, `"random"`, or be assigned a numeric value (see details).

`method` Can take either `"common"` or `"random"` to indicate whether relative effects should be modelled with between-study heterogeneity or not (see details).

`knots` The number/location of knots if a restricted cubic spline dose-response function is fitted (`fun="rcs"`). If a single number is given it indicates the number of knots (they will be equally spaced across the range of doses). If a numeric vector is given it indicates the location of the knots. Minimum number of knots is 3.

`cor` A boolean object that indicates whether correlation should be modelled between relative effect dose-response parameters (`TRUE`) or not (`FALSE`). This is automatically set to `FALSE` if class effects are modelled or if multiple dose-response functions are fitted.

`cor.prior` NOT CURRENTLY IN USE - indicates the prior distribution to use for the correlation/covariance between relative effects. Must be kept as `"wishart"`

`var.scale` A numeric vector indicating the relative scale of variances between correlated dose-response parameters when relative effects are modelled on more than one dose-response parameter and `cor=TRUE` (see details). Each element of the vector refers to the relative scale of each of the dose-response parameters that is modelled using relative effects.

`class.effect` A list of named strings that determines which dose-response parameters to model with a class effect and what that effect should be (`"common"` or `"random"`). Element names should match dose-response parameter names (which will therefore depend on whether or not a wrapper function has been used for `mbnma.run()`). For example: `list("beta.2"="fixed","beta.3"="random")` when using `mbnma.run()` or `list("ed50"="fixed","hill"="random")` when using `mbnma.emax.hill()`.

`UME` A boolean object to indicate whether to fit an Unrelated Mean Effects model that does not assume consistency and so can be used to test if the consistency assumption is valid.

`user.fun` A formula specifying any relationship including `dose` and one/several of: `beta.1`, `beta.2`, `beta.3`, `beta.4`.

`likelihood` A string indicating the likelihood to use in the model. Can take either `"binomial"`, `"normal"` or `"poisson"`. If left as `NULL` the likelihood will be inferred from the data.

`link` A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. `"logit"`, `"log"`, `"probit"`, `"cloglog"`) or be assigned the value `"identity"` for and identity link function. If left as `NULL` the link function will be automatically assigned based on the likelihood.

Details

When relative effects are modelled on more than one dose-response parameter and `cor = TRUE`, correlation between the dose-response parameters is automatically estimated using a vague Wishart prior. This prior can be made slightly more informative by specifying the relative scale of variances between the dose-response parameters using `var.scale`. `cor` will automatically be set to `FALSE` if class effects are modelled or if a model is fitted with multiple dose-response functions.

Value

A single long character string containing the JAGS model generated based on the arguments passed to the function.

Examples

```
# Write model code for a model with an exponential dose-response function,
# relative effects modelled on the rate of growth/decay (beta.1) with a random
# effects model
model <- mbnma.write(fun="exponential",
             beta.1="rel",
             method="random",
             likelihood="binomial",
             link="logit"
             )
cat(model)

# Write model code for a model with an Emax dose-response function,
# relative effects modelled on Emax (beta.1) with a random effects model,
# a single parameter estimated for ED50 (beta.2) with a common effects model
model <- mbnma.write(fun="emax",
             beta.1="rel",
             beta.2="common",
             likelihood="normal",
             link="identity"
             )
cat(model)

# Write model code for a model with an Emax dose-response function,
# relative effects modelled on Emax (beta.1) and ED50 (beta.2).
# Class effects modelled on ED50 with common effects
model <- mbnma.write(fun="emax",
             beta.1="rel",
             beta.2="rel",
             likelihood="normal",
             link="identity",
             class.effect=list("beta.2"="common")
             )
cat(model)

# Write model code for a model with an Emax dose-response function,
# relative effects modelled on Emax (beta.1) and ED50 (beta.2) with a
# random effects model that automatically models a correlation between
```

```
# both parameters.
model <- mbnma.write(fun="emax",
             beta.1="rel",
             beta.2="rel",
             method="random",
             likelihood="normal",
             link="identity",
             )
cat(model)
```

`nma.nodesplit` *Node-splitting model for testing consistency at the treatment level*

Description

Splits contributions for a given set of treatment comparisons into direct and indirect evidence. A discrepancy between the two suggests that the consistency assumption required for NMA and MB-NMA may violated.

Usage

```
nma.nodesplit(
  network,
  likelihood = NULL,
  link = NULL,
  method = "common",
  comparisons = NULL,
  drop.discon = TRUE,
  ...
)

## S3 method for class 'nma.nodesplit'
plot(x, plot.type = NULL, ...)
```

Arguments

`network`	An object of class `mbnma.network`.
`likelihood`	A string indicating the likelihood to use in the model. Can take either `"binomial"`, `"normal"` or `"poisson"`. If left as `NULL` the likelihood will be inferred from the data.
`link`	A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. `"logit"`, `"log"`, `"probit"`, `"cloglog"`) or be assigned the value `"identity"` for and identity link function. If left as `NULL` the link function will be automatically assigned based on the likelihood.
`method`	Can take either `"common"` or `"random"` to indicate whether relative effects should be modelled with between-study heterogeneity or not (see details).

comparisons	A matrix specifying the comparisons to be split (one row per comparison). The matrix must have two columns indicating each treatment for each comparison. Values can either be character (corresponding to the treatment names given in network) or numeric (corresponding to treatment codes within the network - note that these may change if drop.discon = TRUE).
drop.discon	A boolean object that indicates whether to drop treatments that are disconnected at the treatment level. Default is TRUE. If set to FALSE then this could lead to identification of nodesplit comparisons that are not connected to the network reference treatment, or lead to errors in running the nodesplit models, though it can be useful for error checking.
...	Arguments to be sent to ggplot2::ggplot()
x	An object of class("nma.nodesplit")
plot.type	A character string that can take the value of "forest" to plot only forest plots, "density" to plot only density plots, or left as NULL (the default) to plot both types of plot.

Details

The S3 method plot() on an nma.nodesplit object generates either forest plots of posterior medians and 95\% credible intervals, or density plots of posterior densities for direct and indirect evidence.

Value

Plots the desired graph(s) and returns an object (or list of object if plot.type=NULL) of class(c("gg","ggplot"))

Methods (by generic)

- plot: Plot outputs from treatment-level nodesplit models

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

split <- nma.nodesplit(network, likelihood = "binomial", link="logit",
  method="common")



#### To perform nodesplit on selected comparisons ####

# Check for closed loops of treatments with independent evidence sources
loops <- inconsistency.loops(network$data.ab)

# This...
single.split <- nma.nodesplit(network, likelihood = "binomial", link="logit",
             method="random", comparisons=rbind(c("sumatriptan_1", "almotriptan_1")))
```

```
#...is the same as...
single.split <- nma.nodesplit(network, likelihood = "binomial", link="logit",
             method="random", comparisons=rbind(c(6, 12)))


# Plot results
plot(split, plot.type="density") # Plot density plots of posterior densities
plot(split, plot.type="forest") # Plot forest plots of direct and indirect evidence

# Print and summarise results
print(split)
summary(split) # Generate a data frame of summary results
```

`osteopain_2wkabs` *Studies of treatments for pain relief in patients with osteoarthritis*

Description

A dataset from a systematic review of interventions for pain relief in osteoarthritis, used previously in Pedder et al. (2019). The outcome is continuous, and aggregate data responses correspond to the mean WOMAC pain score at 2 weeks follow-up. The dataset includes 18 Randomised-Controlled Trials (RCTs), comparing 8 different agents with placebo. In total there were 26 different treatments (combination of dose and agent). The active treatments can also be grouped into 3 different classes, within which they have similar mechanisms of action.

Usage

```
osteopain_2wkabs
```

Format

A data frame in long format (one row per arm and study), with 74 rows and 7 variables:

- `studyID` Study identifiers
- `agent` Character data indicating the agent to which participants were randomised
- `dose` Numeric data indicating the standardised dose received
- `class` Character data indicating the drug class to which the agent belongs to
- `y` Numeric data indicating the mean pain score on the WOMAC scale in a study arm
- `se` Numeric data indicating the standard error for the mean pain score on the WOMAC scale in a study arm

Source

Pfizer Ltd.

References

Pedder H, Dias S, Bennetts M, Boucher M, Welton NJ (2019). "Modelling time-course relationships with multiple treatments: Model-Based Network Meta-Analysis for continuous summary outcomes." *Res Synth Methods*, **10**(2), 267-286.

`pDcalc` *Calculate plugin pD from a JAGS model with univariate likelihood for studies with repeated measurements*

Description

Uses results from MBNMA JAGS models to calculate pD via the plugin method (Spiegelhalter et al. 2002). Can only be used for models with known standard errors or covariance matrices (typically univariate likelihoods).

Usage

```
pDcalc(
  obs1,
  obs2,
  fups = NULL,
  narm,
  NS,
  theta.result,
  resdev.result,
  likelihood = "normal",
  type = "time"
)
```

Arguments

`obs1`	A matrix (study x arm) or array (study x arm x time point) containing observed data for `y` (normal likelihood) or `r` (binomial or poisson likelihood) in each arm of each study. This will be the same array used as data for the JAGS model.
`obs2`	A matrix (study x arm) or array (study x arm x time point) containing observed data for `se` (normal likelihood), `N` (binomial likelihood) or `E` (poisson likelihood) in each arm of each study. This will be the same array used as data for the JAGS model.
`fups`	A numeric vector of length equal to the number of studies, containing the number of follow-up mean responses reported in each study. Required for time-course MBNMA models (if `type="time"`)
`narm`	A numeric vector of length equal to the number of studies, containing the number of arms in each study.
`NS`	A single number equal to the number of studies in the dataset.

theta.result	A matrix (study x arm) or array (study x arm x time point) containing the posterior mean predicted means/probabilities/rate in each arm of each study. This will be estimated by the JAGS model.
resdev.result	A matrix (study x arm) or array (study x arm x time point) containing the posterior mean residual deviance contributions in each arm of each study. This will be estimated by the JAGS model.
likelihood	A character object of any of the following likelihoods: • normal • binomial (does not work with time-course MBNMA models) • poisson (does not work with time-course MBNMA models)
type	The type of MBNMA model fitted. Can be either "time" or "dose"

Details

Method for calculating pD via the plugin method proposed by Spiegelhalter (Spiegelhalter et al. 2002). Standard errors / covariance matrices must be assumed to be known. To obtain values for theta.result and resdev.result these parameters must be monitored when running the MBNMA model (using parameters.to.save).

For non-linear time-course MBNMA models residual deviance contributions may be skewed, which can lead to non-sensical results when calculating pD via the plugin method. Alternative approaches are to use pV as an approximation or pD calculated by Kullback-Leibler divergence (Plummer 2008).

Value

A single numeric value for pD calculated via the plugin method.

References

Plummer M (2008). "Penalized loss functions for Bayesian model comparison." *Biostatistics*, **9**(3), 523-39. ISSN 1468-4357 (Electronic) 1465-4644 (Linking), https://pubmed.ncbi.nlm.nih.gov/18209015/.

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A (2002). "Bayesian measures of model complexity and fit." *J R Statistic Soc B*, **64**(4), 583-639.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Fit a dose-response MBNMA, monitoring "psi" and "resdev"
result <- mbnma.run(network, fun="exponential", beta.1="rel", method="random",
              parameters.to.save=c("psi", "resdev"))


#### Calculate pD for binomial data ####
```

```
# Prepare data for pD calculation
r <- result$model$data()$r
N <- result$model$data()$N
narm <- result$model$data()$narm
NS <- result$model$data()$NS

psi <- result$BUGSoutput$median$psi
resdevs <- result$BUGSoutput$median$resdev

# Calculate pD via plugin method
pD <- pDcalc(obs1=r, obs2=N, narm=narm, NS=NS,
          theta.result=psi, resdev.result=resdevs,
          likelihood="binomial", type="dose")
```

`plot.mbnma` *Forest plot for results from dose-response MBNMA models*

Description

Generates a forest plot for dose-response parameters.

Usage

```
## S3 method for class 'mbnma'
plot(x, params = NULL, agent.labs = NULL, class.labs = NULL, ...)
```

Arguments

`x`	An S3 object of class `"mbnma"` generated by running a dose-response MBNMA model
`params`	A character vector of dose-response parameters to plot. Parameters must be given the same name as monitored nodes in `mbnma` and must be modelled as relative effects (`"rel"`). Can be set to `NULL` to include all available dose-response parameters estimated by `mbnma`.
`agent.labs`	A character vector of agent labels (including `"Placebo"` if it has been included in the original network). If left as `NULL` (the default) then labels will be used as defined in the data.
`class.labs`	A character vector of class labels if `mbnma` was modelled using class effects (including `"Placebo"` if it has been included in the original network). If left as `NULL` (the default) then labels will be used as defined in the data.
`...`	Arguments to be passed to methods, such as graphical parameters

Value

A forest plot of class `c("gg","ggplot")` that has separate panels for different dose-response parameters. Results are plotted on the link scale.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Run an exponential dose-response MBNMA and generate the forest plot
exponential <- mbnma.run(network, fun="exponential")
plot(exponential)

# Plot only Emax parameters from an Emax dose-response MBNMA
emax <- mbnma.emax(network, emax="rel", ed50="rel", method="random")
plot(emax, params=c("d.emax"))


#### Forest plots including class effects ####
# Generate some classes for the data
class.df <- HF2PPITT
class.df$class <- ifelse(class.df$agent=="placebo", "placebo", "active1")
class.df$class <- ifelse(class.df$agent=="eletriptan", "active2", class.df$class)
netclass <- mbnma.network(class.df)
emax <- mbnma.emax(netclass, emax="rel", ed50="rel", method="random",
            class.effect=list("ed50"="common"))

# Plot forest plot with different labels for classes
plot(emax, class.labs=c("Placebo", "Other Active", "Eletriptan"))

# Since "Placebo" is included in the network, it must be included in labels
# Failure to do so will cause an error
## ERROR ## plot(emax, class.labs=c("Other Active", "Eletriptan"))
```

`plot.mbnma.network` *Create an mbnma.network object*

Description

Creates an object of class `mbnma.network`. Various MBNMA functions can subsequently be applied to this object.

Usage

```
## S3 method for class 'mbnma.network'
plot(
  x,
  level = "treatment",
  v.color = "connect",
  doselink = NULL,
  layout = igraph::in_circle(),
```

```
  remove.loops = FALSE,
  edge.scale = 1,
  v.scale = NULL,
  label.distance = 0,
  legend = TRUE,
  legend.x = "bottomleft",
  legend.y = NULL,
  ...
)

mbnma.network(data.ab, description = "Network")
```

Arguments

x	An object of class mbnma.network.
level	A string indicating whether nodes/facets should represent "treatment" or "agent" in the plot. Can be used to examine the expected impact of modelling dose-response in terms of network connectivity.
v.color	Can take either "connect" (the default) to indicate that nodes should only be coloured if they are connected to the network reference treatment (indicates network connectivity) or "agent" to colour nodes by agent.
doselink	If given an integer value it indicates that connections via the dose-response relationship with placebo should be plotted. The integer represents the minimum number of doses from which a dose-response function could be estimated and is equivalent to the number of parameters in the desired dose-response function plus one. If left as NULL (the default), connections to placebo via dose-response relationships will not be included.
layout	An igraph layout specification. This is a function specifying an igraph layout that determines the arrangement of the vertices (nodes). The default igraph::as_circle() arranged vertices in a circle. Two other useful layouts for network plots are: igraph::as_star(), igraph::with_fr(). Others can be found in layout_
remove.loops	A boolean value indicating whether to include loops that indicate comparisons within a node.
edge.scale	A number to scale the thickness of connecting lines (edges). Line thickness is proportional to the number of studies for a given comparison. Set to 0 to make thickness equal for all comparisons.
v.scale	A number with which to scale the size of the nodes. If the variable N (to indicate the numbers of participants in each study arm) is included in the dataset then the size of the nodes will be proportional to the number of participants within a treatment/agent in the network.
label.distance	A number scaling the distance of labels from the nodes to improve readability. The labels will be directly on top of the nodes if the default of 0 is used. Option only applicable if layout_in_circle is set to TRUE.
legend	A boolean object to indicate whether or not to plot a legend to indicate which node colour corresponds to which agent if v.color="agent". Default is TRUE.

`legend.x, legend.y`
: The x and y co-ordinates to be used to position the legend. They can be specified by keyword or in any way which is accepted by `xy.coords`.

`...`
: Options for plotting in `igraph`.

`data.ab`
: A data frame of arm-level data in "long" format containing the columns:

- `studyID` Study identifiers
- `dose` Numeric data indicating the dose (must take positive values)
- `agent` Agent identifiers (can be numeric, factor or character)
- `y` Numeric data indicating the aggregate response for a continuous outcome. Required for continuous data.
- `se` Numeric data indicating the standard error for a given observation. Required for continuous data.
- `r` Numeric data indicating the number of responders within a study arm. Required for binomial or poisson data.
- `N` Numeric data indicating the total number of participants within a study arm. Required for binomial data
- `E` Numeric data indicating the total exposure time for participants within a study arm. Required for poisson data.
- `class` An optional column indicating a particular class code. Agents with the same identifier must also have the same class code.

`description`
: Optional. Short description of the network.

Details

The S3 method `plot()` on an `mbnma.network` object generates a network plot that shows how different treatments are connected within the network via study comparisons. This can be used to identify how direct and indirect evidence are informing different treatment comparisons. Depends on `igraph`.

Agents/classes for arms that have dose = 0 will be relabelled as `"Placebo"`. Missing values (`NA`) cannot be included in the dataset. Single arm studies cannot be included.

Value

`plot()`: An object of `class("igraph")` - any functions from the `igraph` package can be applied to this object to change its characteristics.

`mbnma.network()`: An object of `class("mbnma.network")` which is a list containing:

- `description` A short description of the network
- `data.ab` A data frame containing the arm-level network data (treatment identifiers will have been recoded to a sequential numeric code)
- `agents` A character vector indicating the agent identifiers that correspond to the new agent codes.
- `treatments` A character vector indicating the treatment identifiers that correspond to the new treatment codes.
- `classes` A character vector indicating the class identifiers (if included in the original data) that correspond to the new class codes.

Methods (by generic)

- plot: Generate a network plot

Examples

```
# Create an mbnma.network object from the data
network <- mbnma.network(HF2PPITT)

# Generate a network plot from the data
plot(network)

# Generate a network plot at the agent level that removes loops indicating comparisons
#within a node
plot(network, level="agent", remove.loops=TRUE)

# Generate a network plot at the treatment level that colours nodes by agent
plot(network, v.color="agent", remove.loops=TRUE)

# Generate a network plot that includes connections via the dose-response function
# For a one parameter dose-response function (e.g. exponential)
plot(network, level="treatment", doselink=1, remove.loops=TRUE)

# For a two parameter dose-response function (e.g. Emax)
plot(network, level="treatment", doselink=2, remove.loops=TRUE)

# Arrange network plot in a star with the reference treatment in the centre
plot(network, layout=igraph::as_star(), label.distance=3)

#### Plot a network with no placebo data included ####
# Make data with no placebo
noplac.df <- network$data.ab[network$data.ab$narm>2 & network$data.ab$agent!=1,]
net.noplac <- mbnma.network(noplac.df)

# Plotting network automatically plots connections to Placebo via dose-response
plot(net.noplac)
# Using the triptans headache dataset
print(HF2PPITT)

# Define network
network <- mbnma.network(HF2PPITT, description="Example")

# Plot network
plot(network)
```

---

plot.mbnma.predict *Plots predicted responses from a dose-response MBNMA model*

Description

Plots predicted responses on the natural scale from a dose-response MBNMA model.

Usage

```
## S3 method for class 'mbnma.predict'
plot(
  x,
  disp.obs = FALSE,
  overlay.split = FALSE,
  method = "common",
  agent.labs = NULL,
  scales = "free_x",
  ...
)
```

Arguments

x	An object of class "mbnma.predict" generated by predict("mbnma")
disp.obs	A boolean object to indicate whether to show the location of observed doses in the data on the 95\% credible intervals of the predicted dose-response curves as shaded regions (TRUE) or not (FALSE). If set to TRUE the original network object used for the model **must** be specified in network.
overlay.split	A boolean object indicating whether to overlay a line showing the split (treatment-level) NMA results on the plot (TRUE) or not (FALSE). This will require automatic running of a split NMA model. For overlay.split=TRUE the original network object used for the model **must** be specified in network.
method	Indicates the type of split (treatment-level) NMA to perform when overlay.split=TRUE. Can take either "common" or "random".
agent.labs	A character vector of agent labels to display on plots. If left as NULL (the default) the names of agents will be taken from predict. The position of each label corresponds to each element of predict. The number of labels must equal the number of active agents in predict. If placebo / dose=0 data is included in the predictions then a label for placebo **should not** be included in agent.labs. It will not be shown in the final plot since placebo is the point within each plot at which dose = 0 (rather than a separate agent).
scales	Should scales be fixed ("fixed", the default), free ("free"), or free in one dimension ("free_x", "free_y")?
...	Arguments for ggplot2

Details

For the S3 method plot(), it is advisable to ensure predictions in predict are estimated using a sufficient number of doses to ensure a smooth predicted dose-response curve. If disp.obs = TRUE it is advisable to ensure predictions in predict are estimated using an even sequence of time points to avoid misrepresentation of shaded densities.

Examples

```
# Using the triptans data
```

```
network <- mbnma.network(HF2PPITT)

# Run an Emax dose-response MBNMA and predict responses
emax <- mbnma.emax(network, method="random")
pred <- predict(emax, E0 = 0.5)
plot(pred)

# Display observed doses on the plot
plot(pred, disp.obs=TRUE)

# Display split NMA results on the plot
plot(pred, overlay.split=TRUE)

# Split NMA results estimated using random treatment effects model
plot(pred, overlay.split=TRUE, method="random")

# Add agent labels
plot(pred, agent.labs=c("Elet", "Suma", "Frov", "Almo", "Zolmi",
      "Nara", "Riza"))

# These labels will throw an error because "Placebo" is included in agent.labs when
#it will not be plotted as a separate panel
#### ERROR ####
#plot(pred, agent.labs=c("Placebo", "Elet", "Suma", "Frov", "Almo", "Zolmi",
#      "Nara", "Riza"))


# If insufficient predictions are made across dose-response function
# then the plotted responses are less smooth and can be misleading
pred <- predict(emax, E0 = 0.5, n.doses=3)
plot(pred)
```

`plot.mbnma.rank` *Plot histograms of rankings from MBNMA models*

Description

Plot histograms of rankings from MBNMA models

Usage

```
## S3 method for class 'mbnma.rank'
plot(x, params = NULL, treat.labs = NULL, ...)
```

Arguments

`x`	An object of class `"mbnma.rank"` generated by `rank.mbnma()`

params	A character vector of named parameters in the model that vary by either agent or class (depending on the value assigned to level). If left as NULL (the default), then ranking will be calculated for all available parameters that vary by agent/class.
treat.labs	A vector of treatment labels in the same order as treatment codes. Easiest to use treatment labels stored by mbnma.network()
...	Arguments to be sent to ggplot::geom_bar()

Value

A series of histograms that show rankings for each treatment/agent/prediction, with a separate panel for each parameter The object returned is a list containing a separate element for each parameter in params which is an object of class(c("gg","ggplot")).

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Estimate rankings  from an Emax dose-response MBNMA
emax <- mbnma.emax(network, emax="rel", ed50="rel", method="random")
ranks <- rank(emax)

# Plot rankings for both dose-response parameters (in two separate plots)
plot(ranks)

# Plot rankings just for ED50
plot(ranks, params="d.ed50")

# Plot rankings from prediction
doses <- list("eletriptan"=c(0,1,2,3), "rizatriptan"=c(0.5,1,2))
pred <- predict(emax, E0 = "rbeta(n, shape1=1, shape2=5)",
            exact.doses=doses)
rank <- rank(pred)
plot(rank)
```

---

plot.nma *Run an NMA model*

Description

Used for calculating split NMA results, either when comparing models that do not account for dose-response relationship, or to estimate split results for overlay.split. Results can also be compared between consistency (UME=FALSE) and inconsistency (UME=TRUE) models to test the validity of the consistency assumption.

Usage

```
## S3 method for class 'nma'
plot(x, bydose = TRUE, scales = "free_x", ...)

nma.run(
  network,
  method = "common",
  likelihood = NULL,
  link = NULL,
  priors = NULL,
  warn.rhat = TRUE,
  n.iter = 10000,
  drop.discon = TRUE,
  UME = FALSE,
  pd = "pv",
  ...
)
```

Arguments

x	An object of class("nma")
bydose	A boolean object indicating whether to plot responses with dose on the x-axis (TRUE) to be able to examine potential dose-response shapes, or to plot a conventional forest plot with all treatments on the same plot (FALSE)
scales	Should scales be fixed ("fixed", the default), free ("free"), or free in one dimension ("free_x", "free_y")?
...	Arguments to be sent to ggplot2::ggplot()
network	An object of class mbnma.network.
method	Indicates the type of split (treatment-level) NMA to perform when overlay.split=TRUE. Can take either "common" or "random".
likelihood	A string indicating the likelihood to use in the model. Can take either "binomial", "normal" or "poisson". If left as NULL the likelihood will be inferred from the data.
link	A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. "logit", "log", "probit", "cloglog") or be assigned the value "identity" for and identity link function. If left as NULL the link function will be automatically assigned based on the likelihood.
priors	A named list of parameter values (without indices) and replacement prior distribution values given as strings **using distributions as specified in JAGS syntax** (see examples).
warn.rhat	A boolean object to indicate whether to return a warning if Rhat values for any monitored parameter are >1.02 (suggestive of non-convergence).
n.iter	number of total iterations per chain (including burn in; default: 10000)
drop.discon	A boolean object that indicates whether or not to drop disconnected studies from the network.

UME — A boolean object to indicate whether to fit an Unrelated Mean Effects model that does not assume consistency and so can be used to test if the consistency assumption is valid.

pd — Can take either:

- pv only pV will be reported (as automatically outputted by R2jags).
- plugin calculates pD by the plug-in method (Spiegelhalter et al. 2002). It is faster, but may output negative non-sensical values, due to skewed deviances that can arise with non-linear models.
- pd.kl calculates pD by the Kullback-Leibler divergence (Plummer 2008). This will require running the model for additional iterations but will always produce a positive result.
- popt calculates pD using an optimism adjustment which allows for calculation of the penalized expected deviance (Plummer 2008)

Methods (by generic)

- plot: Plot outputs from treatment-level NMA models
  Results can be plotted either as a single forest plot, or facetted by agent and plotted with increasing dose in order to identify potential dose-response relationships.

Examples

```
# Run random effects NMA on the alogliptin dataset
network <- mbnma.network(alog_pcfb)
nma <- nma.run(network, method="random")
print(nma)
plot(nma)

# Run common effects NMA keeping treatments that are disconnected in the NMA
network <- mbnma.network(GoutSUA_2wkCFB)
nma <- nma.run(network, method="common", drop.discon=FALSE)

# Run an Unrelated Mean Effects (UME) inconsistency model on triptans dataset
network <- mbnma.network(HF2PPITT)
ume <- nma.run(network, method="random", UME=TRUE)
```

---

predict.mbnma — *Predict responses for different doses of agents in a given population based on MBNMA dose-response models*

Description

Used to predict responses for different doses of agents or to predict the results of a new study. This is calculated by combining relative treatment effects with a given reference treatment response (specific to the population of interest).

Usage

```
## S3 method for class 'mbnma'
predict(
  object,
  n.doses = 15,
  max.doses = NULL,
  exact.doses = NULL,
  E0 = 0,
  synth = "fixed",
  ...
)
```

Arguments

object	An S3 object of class "mbnma" generated by running a dose-response MBNMA model
n.doses	A number indicating the number of doses at which to make predictions within each agent. The default is 15.
max.doses	A list of numbers. Each named element in the list corresponds to an agent (either named similarly to agent names given in the data, or named correspondingly to the codes for agents given in mbnma) and each number for that element corresponds to the maximum dose of the given agent, below which several predictions will be calculated at different doses (the number of these is determined by n.doses). Can only take positive values. If left as NULL (the default) results will be predicted based on the maximum dose of each agent given in the data.
exact.doses	A list of numeric vectors. Each named element in the list correponds to an agent (either named similarly to agent names given in the data, or named correspondingly to the codes for agents given in mbnma) and each number within the vector for that element corresponds to a dose of the agent for which to predict responses. Doses can only take positive values.
E0	An object to indicate the value(s) to use for the response at dose = 0 (i.e. placebo) in the prediction. This can take a number of different formats depending on how it will be used/calculated. The default is 0 but this will typically lead to nonsensical predictions. • numeric() A single numeric value representing the deterministic response at dose = 0, given on the natural scale - so for binomial data, proportions should be given and for Poisson data, a rate should be given. • character() A single string representing a stochastic distribution for the response at dose = 0, given on the natural scale - so for binomial data, proportions should be given and for Poisson data, a rate should be given. This is specified as a random number generator (RNG) given as a string, and can take any RNG distribution for which a function exists in R. For example: "rnorm(n,7,0.5)". • data.frame() A data frame containing data in the long format (one row per study arm) to be meta-analysed to estimate the dose = 0 (placebo) response.

This could be a set of observational studies that are specific to the population on which to make predictions, or it can be a subset of the study arms within the MBNMA dataset that investigate placebo. See `ref.synth()`

`synth` A character object that can take the value `"fixed"` or `"random"` to specify the the type of pooling to use for synthesis of `E0` if a data frame has been provided for it. Using `"random"` rather than `"fixed"` for `synth` will result in wider 95\% CrI for predictions.

`...` Arguments to be sent to `R2jags::jags()` for synthesis of the network reference treatment effect (using `ref.synth()`)

Details

The range of doses on which to make predictions can be specified in one of two ways:

1. Use `max.dose` and `n.doses` to specify the maximum dose for each agent and the number of doses within that agent for which to predict responses. Doses will be chosen that are equally spaced from zero to the maximum dose for each agent. This is useful for generating plots of predicted responses (using [plot-mbnma.predict]) as it will lead to fitting a smooth dose-response curve (provided `n.doses` is sufficiently high).
2. Use `exact.doses` to specify the exact doses for which to predict responses for each agent. This may be more useful when ranking different predicted responses using [rank-mbnma.predict]

Value

An S3 object of class `mbnma.predict` that contains the following elements:

- `summary` A named list of data frames. Each data frame contains a summary of predicted responses at follow-up times specified in `times` for each treatment specified in `treats`
- `pred.mat` A named list of matrices. Each matrix contains the MCMC results of predicted responses at follow-up times specified in `times` for each treatment specified in `treats`

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Run an Emax dose-response MBNMA
emax <- mbnma.emax(network, emax="rel", ed50="rel", method="random")


############################
###### Specifying E0 ######
############################

#### Predict responses using deterministic value for E0 ####
# Data is binomial so we specify E0 on the natural scale as a probability
pred <- predict(emax, E0 = 0.2)

# Specifying non-sensical values will return an error
```

```
#pred <- predict(emax, E0 = -10)
### ERROR ###

#### Predict responses using stochastic value for E0 ####
# Data is binomial so we might want to draw from a beta distribution
pred <- predict(emax, E0 = "rbeta(n, shape1=1, shape2=5)")

# Misspecifying the RNG string will return an error
#pred <- predict(emax, E0 = "rbeta(shape1=1, shape2=5)")
### ERROR ###


#### Predict responses using meta-analysis of dose = 0 studies ####

# E0 is assigned a data frame of studies to synthesis
# Can be taken from placebo arms in triptans dataset
ref.df <- network$data.ab[network$data.ab$agent==1,]

# Synthesis can be fixed/random effects
pred <- predict(emax, E0 = ref.df, synth="random")



#########################################################################
#### Specifying which doses/agents for which to predict responses ####
#########################################################################

# Change the number of predictions for each agent
pred <- predict(emax, E0 = 0.2, n.doses=20)
pred <- predict(emax, E0 = 0.2, n.doses=3)

# Change the range of predicted doses to be the same for all agents
# But only predict responses for a subset of agents
pred <- predict(emax, E0 = 0.2,
            max.doses=list("Placebo"=0, "eletriptan"=5, "sumatriptan"=5))
plot(pred) # Plot predictions

# Specify several exact combinations of doses and agents to predict
pred <- predict(emax, E0 = 0.2,
            exact.doses=list("eletriptan"=c(0:5), "sumatriptan"=c(1,3,5)))
plot(pred) # Plot predictions

# Print and summarise `mbnma.predict` object
print(pred)
summary(pred)

# Plot `mbnma.predict` object
plot(pred)
```

print.mbnma.network *Print mbnma.network information to the console*

Description

Print mbnma.network information to the console

Usage

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

Arguments

x	An object of class mbnma.network.
...	further arguments passed to or from other methods

print.mbnma.predict *Print summary information from an mbnma.predict object*

Description

Print summary information from an mbnma.predict object

Usage

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

Arguments

x	An object of class("mbnma.predict") generated by predict.mbnma()
...	further arguments passed to or from other methods

print.mbnma.rank *Prints summary information about an mbnma.rank object*

Description

Prints summary information about an mbnma.rank object

Usage

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

Arguments

x	An object of class "mbnma.rank" generated by rank.mbnma()
...	further arguments passed to or from other methods

print.nma.nodesplit *Prints summary results from an nma.nodesplit object*

Description

Prints summary results from an nma.nodesplit object

Usage

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

Arguments

x	An object of class("nma.nodesplit")
...	further arguments passed to or from other methods

print.nodesplit *Prints summary results from a nodesplit object*

Description

Prints summary results from a nodesplit object

Usage

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

Arguments

x	An object of class("nodesplit")
...	further arguments passed to or from other methods

psoriasis *Studies of biologics for treatment of moderate-to-severe psoriasis*

Description

A dataset from a systematic review of Randomised-Controlled Trials (RCTs) comparing biologics at different doses and placebo (Warren et al. 2019). Three different binary outcomes are included, all based on the number of patients experiencing degrees of improvement on the Psoriasis Area and Severity Index (PASI) measured at 12 weeks follow-up. The dataset includes 28 Randomised-Controlled Trials (RCTs), comparing 9 different biologics at different doses with placebo.

Usage

```
psoriasis
```

Format

A data frame in long format (one row per arm and study), with 81 rows and 9 variables:

- studyID Study identifiers
- agent Character data indicating the agent to which participants were randomised
- dose_mg Numeric data indicating the dose to which participants were randomised in mg
- freq Character data indicating the frequency of the dose to which participants were randomised
- dose Numeric data indicating the dose in mg/week to which the participants were randomised
- N Numeric data indicating the number of participants randomised

- `r75` Numeric data indicating the number of participants who achieved >=75% improvement in PASI score after 12 weeks
- `r90` Numeric data indicating the number of participants who achieved >=90% improvement in PASI score after 12 weeks
- `r100` Numeric data indicating the number of participants who achieved 100% improvement in PASI score after 12 weeks

References

Warren RB, Gooderham M, Burge R, Zhu B, Amato D, Liu KH, Shrom D, Guo J, Brnabic A, Blauvelt A (2019). "Comparison of cumulative clinical benefits of biologics for the treatment of psoriasis over 16 weeks: Results from a network meta-analysis." *J Am Acad Dermatol*, **82**(5), 1138-1149.

`rank` *Set rank as a method*

Description

Set rank as a method

Usage

```
rank(x, ...)
```

Arguments

`x`	An object on which to apply the rank method
`...`	Arguments to be passed to methods

`rank.mbnma` *Rank parameter estimates*

Description

Only parameters that vary by agent/class can be ranked.

Usage

```
## S3 method for class 'mbnma'
rank(x, params = NULL, direction = 1, level = "agent", to.rank = NULL, ...)
```

Arguments

x	An object on which to apply the rank method
params	A character vector of named parameters in the model that vary by either agent or class (depending on the value assigned to level). If left as NULL (the default), then ranking will be calculated for all available parameters that vary by agent/class.
direction	Indicates whether negative responses are better (taking the value -1) or positive responses are better (taking the value 1)
level	Can be set to "agent" to rank across different agents or "class" to rank across different classes.
to.rank	A numeric vector containing the codes for the agents/classes you wish to rank. If left NULL then all agents/classes (depending on the value assigned to level) in the model will be ranked. Included codes must be greater than 2 if placebo has been modelled, since placebo cannot be included in the ranking
...	Arguments to be passed to methods

Details

Ranking cannot currently be performed on non-parametric dose-response MBNMA

Value

An object of class("mbnma.rank") which is a list containing a summary data frame, a matrix of rankings for each MCMC iteration, a matrix of probabilities that each agent has a particular rank, and a matrix of cumulative ranking probabilities for each agent, for each parameter that has been ranked.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Rank selected agents from a linear dose-response MBNMA
linear <- mbnma.run(network, fun="linear")
ranks <- rank(linear, to.rank=c("zolmitriptan", "eletriptan", "sumatriptan"))
summary(ranks)

# Rank only ED50 parameters from an Emax dose-response MBNMA
emax <- mbnma.emax(network, emax="rel", ed50="rel", method="random")
ranks <- rank(emax, params="d.ed50")
summary(ranks)


#### Ranking by class ####
# Generate some classes for the data
class.df <- HF2PPITT
class.df$class <- ifelse(class.df$agent=="placebo", "placebo", "active1")
class.df$class <- ifelse(class.df$agent=="eletriptan", "active2", class.df$class)
```

```
netclass <- mbnma.network(class.df)
emax <- mbnma.emax(netclass, emax="rel", ed50="rel", method="random",
            class.effect=list("ed50"="common"))

# Rank by class, with negative responses being "better"
ranks <- rank(emax, level="class", direction=-1)
print(ranks)

# Print and generate summary data frame for `mbnma.rank` object
summary(ranks)
print(ranks)

# Plot `mbnma.rank` object
plot(ranks)
```

`rank.mbnma.predict` *Rank predicted doses of different agents*

Description

Rank predicted doses of different agents

Usage

```
## S3 method for class 'mbnma.predict'
rank(x, direction = 1, rank.doses = NULL, ...)
```

Arguments

`x`	An object on which to apply the rank method
`direction`	Indicates whether negative responses are better (taking the value `-1`) or positive responses are better (taking the value `1`)
`rank.doses`	A list of numeric vectors. Each named element corresponds to an agent (as named/coded in `predict`), and each number within the vector for that element corresponds to the dose for that agent. Doses of agents specified in `rank.doses` **must** be a subset of those for which responses have been predicted in `predict`. If left as `NULL` (the default) then all doses of all agents in `predict` will be ranked.
`...`	Arguments to be passed to methods

Details

If `predict` contains multiple predictions at dose=0, then only the first of these will be included, to avoid duplicating rankings.

Value

An object of `class("mbnma.rank")` which is a list containing a summary data frame, a matrix of rankings for each MCMC iteration, and a matrix of probabilities that each agent has a particular rank, for each parameter that has been ranked.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Rank predictions from a linear dose-response MBNMA
linear <- mbnma.run(network, fun="linear")
pred <- predict(linear, E0 = 0.5)
rank <- rank(pred)
summary(rank)

# Rank selected predictions from an Emax dose-response MBNMA
emax <- mbnma.emax(network, emax="rel", ed50="rel", method="random")
doses <- list("eletriptan"=c(0,1,2,3), "rizatriptan"=c(0.5,1,2))
pred <- predict(emax, E0 = "rbeta(n, shape1=1, shape2=5)",
            exact.doses=doses)
rank <- rank(pred,
            rank.doses=list("eletriptan"=c(0,2), "rizatriptan"=2))

# Print and generate summary data frame for `mbnma.rank` object
summary(rank)
print(rank)

# Plot `mbnma.rank` object
plot(rank)
```

---

ref.synth *Synthesise single arm dose = 0 / placebo studies to estimate E0*

Description

Synthesises single arm studies to estimate E0. Used in predicting responses from a dose-response MBNMA.

Usage

```
ref.synth(
  data.ab,
  mbnma,
  synth = "fixed",
  n.iter = mbnma$BUGSoutput$n.iter,
```

```
  n.burnin = mbnma$BUGSoutput$n.burnin,
  n.thin = mbnma$BUGSoutput$n.thin,
  n.chains = mbnma$BUGSoutput$n.chains,
  ...
)
```

Arguments

`data.ab`	A data frame of arm-level data in "long" format containing the columns: • `studyID` Study identifiers • `y` Numeric data indicating the aggregate response for a continuous outcome. Required for continuous data. • `se` Numeric data indicating the standard error for a given observation. Required for continuous data. • `r` Numeric data indicating the number of responders within a study arm. Required for binomial or poisson data. • `N` Numeric data indicating the total number of participants within a study arm. Required for binomial data • `E` Numeric data indicating the total exposure time for participants within a study arm. Required for poisson data.
`mbnma`	An S3 object of class `"mbnma"` generated by running a dose-response MBNMA model
`synth`	A character object that can take the value `"fixed"` or `"random"` to specify the the type of pooling to use for synthesis of `E0` if a data frame has been provided for it. Using `"random"` rather than `"fixed"` for `synth` will result in wider 95\% CrI for predictions.
`n.iter`	number of total iterations per chain (including burn in; default: 2000)
`n.burnin`	length of burn in, i.e. number of iterations to discard at the beginning. Default is `n.iter/2`, that is, discarding the first half of the simulations. If n.burnin is 0, `jags()` will run 100 iterations for adaption.
`n.thin`	thinning rate. Must be a positive integer. Set `n.thin` > 1 to save memory and computation time if `n.iter` is large. Default is `max(1,floor(n.chains * (n.iter-n.burnin) / 1000))` which will only thin if there are at least 2000 simulations.
`n.chains`	number of Markov chains (default: 3)
`...`	Arguments to be sent to `R2jags::jags()` for synthesis of the network reference treatment effect (using `ref.synth()`)

Details

`data.ab` can be a collection of studies that closely resemble the population of interest intended for the prediction, which could be different to those used to estimate the MBNMA model, and could include single arms of RCTs or observational studies. If other data is not available, the data used to estimate the MBNMA model can be used by selecting only the studies and arms that investigate dose = 0 (placebo).

Defaults for `n.iter`, `n.burnin`, `n.thin` and `n.chains` are those used to estimate `mbnma`.

Value

A list of named elements corresponding to E0 and the between-study standard deviation for E0 if `synth="random"`. Each element contains the full MCMC results from the synthesis.

Examples

```
# Using the triptans data
network <- mbnma.network(HF2PPITT)

# Run an Emax dose-response MBNMA
emax <- mbnma.emax(network, emax="rel", ed50="rel", method="random")

# Data frame for synthesis can be taken from placebo arms
ref.df <- HF2PPITT[HF2PPITT$agent=="placebo",]

# Meta-analyse placebo studies using fixed treatment effects
E0 <- ref.synth(ref.df, emax, synth="fixed")
names(E0)

# Meta-analyse placebo studies using random treatment effects
E0 <- ref.synth(ref.df, emax, synth="random")
names(E0)
```

---

rescale.link *Rescale data depending on the link function provided*

Description

Rescale data depending on the link function provided

Usage

```
rescale.link(x, direction = "link", link = "logit")
```

Arguments

x	A numeric vector of data to be rescaled
direction	Can take either `"link"` to convert data to a particular scale as defined by the `link` function, or `"natural"` to return it to the natural scale.
link	A string indicating the link function to use in the model. Can take any link function defined within JAGS (e.g. `"logit"`, `"log"`, `"probit"`, `"cloglog"`) or be assigned the value `"identity"` for and identity link function. If left as `NULL` the link function will be automatically assigned based on the likelihood.

Value

A rescaled numeric vector

`ssri` *Studies of Selective Serotonin Reuptake Inhibitors (SSRIs) for major depression*

Description

A dataset from a systematic review examining the efficacy of different doses of SSRI antidepressant drugs and placebo (Furukawa et al. 2019). The response to treatment is defined as a 50% reduction in depressive symptoms after 8 weeks (4-12 week range) follow-up. The dataset includes 60 RCTs comparing 5 different SSRIs with placebo.

Usage

```
ssri
```

Format

A data frame in long format (one row per arm and study), with 145 rows and 8 variables:

- `studyID` Study identifiers
- `bias` Risk of bias evaluated on 6 domains
- `age` Mean participant age
- `weeks` Duration of study follow-up
- `agent` Character data indicating the agent to which participants were randomised
- `dose` Numeric data indicating the dose to which participants were randomised in mg
- `N` Numeric data indicating the number of participants randomised
- `r` Numeric data indicating the number of participants who achieved >50% improvement in depression symptoms

References

Furukawa TA, Cipriani A, Cowen PJ, Leucht S, Egger M, Salanti G (2019). "Optimal dose of selective serotonin reuptake inhibitors, venlafaxine, and mirtazapine in major depression: a systematic review and dose-response meta-analysis." *Lancet Psychiatry*, **6**, 601-609.

summary.mbnma *Print summary of MBNMA results to the console*

Description

Print summary of MBNMA results to the console

Usage

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

Arguments

object	An S3 object of class "mbnma" generated by running a dose-response MBNMA model
...	additional arguments affecting the summary produced

summary.mbnma.network *Print summary mbnma.network information to the console*

Description

Print summary mbnma.network information to the console

Usage

```
## S3 method for class 'mbnma.network'
summary(object, ...)
```

Arguments

object	An object of class mbnma.network.
...	further arguments passed to or from other methods

summary.mbnma.predict *Produces a summary data frame from an mbnma.predict object*

Description

Produces a summary data frame from an mbnma.predict object

Usage

```
## S3 method for class 'mbnma.predict'
summary(object, ...)
```

Arguments

object	An object of class("mbnma.predict)" generated by predict("mbnma")
...	additional arguments affecting the summary produced.

Value

A data frame containing posterior summary statistics from predicted responses from a dose-response MBNMA model

summary.mbnma.rank *Generates summary data frames for an mbnma.rank object*

Description

Generates summary data frames for an mbnma.rank object

Usage

```
## S3 method for class 'mbnma.rank'
summary(object, ...)
```

Arguments

object	An object of class("mbnma.rank") generated by rank.mbnma()
...	additional arguments affecting the summary produced

Value

A list in which each element represents a parameter that has been ranked in mbnma.rank and contains a data frame of summary ranking results.

summary.nma.nodesplit *Generates a summary data frame for nma.nodesplit objects*

Description

Generates a summary data frame for nma.nodesplit objects

Usage

```
## S3 method for class 'nma.nodesplit'
summary(object, ...)
```

Arguments

object	An object of class("nma.nodesplit")
...	further arguments passed to or from other methods

summary.nodesplit *Generates a summary data frame for nodesplit objects*

Description

Generates a summary data frame for nodesplit objects

Usage

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

Arguments

object	An object of class("nodesplit")
...	further arguments passed to or from other methods

Index