Package: MBNMAdose (via r-universe)

October 24, 2026

Type Package

Title Dose-Response MBNMA Models

Version 0.5.1

Language en-GB

Date 2026-06-24

URL https://hugaped.github.io/MBNMAdose/

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, methods, scales, dplyr (>= 0.7.4), R2jags (>= 0.5-7), rjags (>= 4-8), magrittr (>= 1.5), checkmate (>= 1.8.5), Rdpack (>= 0.11-0), igraph (>= 2.0.1.1), ggplot2 (>= 2.2.1), overlapping (>= 1.5.0), reshape2 (>= 1.4.3)

Suggests RColorBrewer (>= 1.1-2), coda (>= 0.19-4), testthat (>= 1.0.2), crayon (>= 1.3.4), ggdist (>= 2.4.0), zoo (>= 1.8-8), formatR (>= 1.14), netmeta, knitr, rmarkdown

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

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RdMacros Rdpack

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Hugo Pedder [aut, cre] (ORCID:
<https://orcid.org/0000-0002-7813-3749>), Adil Karim [ctb]

Config/pak/sysreqs jags

Repository https://cran.r-universe.dev

Date/Publication 2026-06-27 15:20:02 UTC

RemoteUrl https://github.com/cran/MBNMAdose

RemoteRef HEAD

RemoteSha 3e3db8478a43325aaa1fb3c4d62b14a0333e3adc

Contents

add_index 4
alog_pcfb 5
calc.edx 6
check.network 6
cumrank 7
default.priors 8
demax 9
devdev 10
devplot 11
dexp 13
dfpoly 14
ditp 16
dloglin 17
dmulti 18
dnonparam 18
dpoly 19
DR.comparisons 21
drop.comp 21
drop.disconnected 22
dspline 23
duser 24
fitplot 26
gen.parameters.to.save 27
genspline 28
get.prior 29
get.relative 30
getjagsdata 32
gout 35

inconsistency.loops 35
mbnma.comparisons 37
mbnma.nodesplit 38
mbnma.run 40
mbnma.update 47
mbnma.validate.data 48
mbnma.write 49
nma.nodesplit 52
norm2lnorm 54
osteopain 55
pDcalc 56
plot.mbnma 58
plot.mbnma.network 59
plot.mbnma.predict 62
plot.mbnma.rank 64
plot.nma 65
predict.mbnma 68
print.mbnma.network 71
print.mbnma.predict 71
print.mbnma.rank 72
print.nma.nodesplit 72
print.nodesplit 73
print.relative.array 73
psoriasis100 74
psoriasis75 75
psoriasis90 76
rank 77
rank.mbnma 77
rank.mbnma.predict 79
rank.relative.array 80
recode.agent 81
ref.synth 82
rescale.link 84
ssi_closure 85
ssri 85
summary.mbnma 86
summary.mbnma.network 87
summary.mbnma.predict 87
summary.mbnma.rank 88
summary.nma.nodesplit 88
summary.nodesplit 89
triptans 89

Index **91**

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 or when modelling Standardised Mean Differences • 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. • standsd An optional column of numeric data indicating reference SDs used to standardise treatment effects when modelling using Standardised Mean Differences (SMD).
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.

Usage

```
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
- `n` Numeric data indicating the number of participants randomised

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`.

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.

`calc.edx` *Calculates values for EDx from an Emax model, the dose at which x% of the maximal response (Emax) is reached*

Description

Calculates values for EDx from an Emax model, the dose at which x% of the maximal response (Emax) is reached

Usage

```
calc.edx(mbnma, x = 50)
```

Arguments

`mbnma`	An S3 object of class `"mbnma"` generated by running a dose-response MBNMA model
`x`	A numeric value between 0 and 100 for the dose at which x% of the maximal response (Emax) should be calculated

Value

A data frame of posterior EDx summary values for each agent

`check.network` *Check if all nodes in the network are connected (identical to function in* `MBNMAtime`*)*

Description

Check if all nodes in the network are connected (identical to function in `MBNMAtime`)

Usage

```
check.network(g, reference = 1)
```

Arguments

`g`	An network plot of `class("igraph")`
`reference`	A numeric value indicating which treatment code to use as the reference treatment for testing that all other treatments connect to it

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 trapezoid approach.
...	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(triptans)

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

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

`default.priors` *Sets default priors for JAGS model code*

Description

This function creates JAGS code snippets for default MBNMA model priors.

Usage

```
default.priors(
  fun = dloglin(),
  UME = FALSE,
  regress.mat = NULL,
  regress.effect = "common",
  om = list(rel = 5, abs = 10)
)
```

Arguments

`fun`	An object of `class("dosefun")` that specifies a functional form to be assigned to the dose-response. See Details.
`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.
`regress.mat`	A Nstudy x Ncovariate design matrix of meta-regression covariates
`regress.effect`	Indicates whether effect modification should be assumed to be `"common"` (assumed to be equal versus Placebo throughout the network), `"random"` (assumed to be exchangeable versus Placebo throughout the network), `"agent"` (assumed to be equal versus Placebo within each agent), or `"class"` (assumed to be equal versus Placebo within each class).
`om`	a list with two elements that report the maximum relative (`"rel"`) and maximum absolute (`"abs"`) efficacies on the link scale.

Value

A list, each element of which is a named JAGS snippet corresponding to a prior in the MBNMA JAGS code.

Examples

```
default.priors(fun=demax())
```

demax *Emax dose-response function*

Description

Emax dose-response function

Usage

```
demax(emax = "rel", ed50 = "rel", hill = NULL, p.expon = FALSE)
```

Arguments

emax	Pooling for Emax parameter. Can take "rel", "common", "random" or be assigned a numeric value (see details).
ed50	Pooling for ED50 parameter. Can take "rel", "common", "random" or be assigned a numeric value (see details).
hill	Pooling for Hill parameter. Can take "rel", "common", "random" or be assigned a numeric value (see details).
p.expon	A logical object to indicate whether ed50 and hill parameters should be expressed within the dose-response function on an exponential scale

Details

Emax represents the maximum response. exp(ED50) represents the dose at which 50% of the maximum response is achieved. exp(Hill) is the Hill parameter, which allows for a sigmoidal function.

Without Hill parameter:

$$\frac{E_{max} \times x}{ET_{50} + x}$$

With Hill parameter:

$$\frac{E_{max} \times x^{hill}}{ET_{50} \times hill} + x^{hill}$$

Value

An object of class("dosefun")

Dose-response parameters

Argument	Model specification
"rel"	Implies that *relative* effects should be pooled for this dose-response parameter separately for each agent in the n
"common"	Implies that all agents share the same common effect for this dose-response parameter.
"random"	Implies that all agents share a similar (exchangeable) effect for this dose-response parameter. This approach allo
numeric()	Assigned a numeric value, indicating that this dose-response parameter should not be estimated from the data bu

When relative effects are modelled on more than one dose-response parameter, correlation between them is automatically estimated using a vague inverse-Wishart prior. This prior can be made slightly more informative by specifying the scale matrix `omega` and by changing the degrees of freedom of the inverse-Wishart prior using the `priors` argument in `mbnma.run()`.

References

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

Examples

```
# Model without a Hill parameter
demax(emax="rel", ed50="common")

# Model including a Hill parameter and defaults for Emax and ED50 parameters
demax(hill="common")
```

`devdev` *Dev-dev plot for comparing deviance contributions from two models*

Description

Plots the deviances of two model types for comparison. Often used to assess consistency by comparing consistency (NMA or MBNMA) and unrelated mean effects (UME) models (see Pedder et al. (2021)). Models must be run on the *same set of data* or the deviance comparisons will not be valid.

Usage

```
devdev(mod1, mod2, dev.type = "resdev", n.iter = 2000, n.thin = 1, ...)
```

Arguments

`mod1`	First model for which to plot deviance contributions
`mod2`	Second model for which to plot deviance contributions
`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`

Examples

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

# Run an poorly fitting linear dose-response
lin <- mbnma.run(network, fun=dpoly(degree=1))

# Run a better fitting Emax dose-response
emax <- mbnma.run(network, fun=demax())

# Run a standard NMA with unrelated mean effects (UME)
ume <- nma.run(network, UME=TRUE)

# Compare residual deviance contributions from linear and Emax
devdev(lin, emax) # Suggests model fit is very different

# Compare deviance contributions from Emax and UME
devdev(emax, ume) # Suggests model fit is similar
```

---

devplot *Plot deviance contributions from an MBNMA model*

Description

Plot deviance contributions from an MBNMA model

Usage

```
devplot(
  mbnma,
  plot.type = "box",
  facet = TRUE,
  dev.type = "resdev",
  n.iter = mbnma$BUGSoutput$n.iter/2,
  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") 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(triptans)

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

# Plot deviances
devplot(emax)

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

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

# 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")
```

---

dexp *Exponential dose-response function*

Description

Similar parameterisation to the Emax model but with non-asymptotic maximal effect (Emax). Can fit a 1-parameter (Emax only) or 2-parameter model (includes onset parameter that controls the curvature of the dose-response relationship)

Usage

```
dexp(emax = "rel", onset = NULL, p.expon = FALSE)
```

Arguments

emax	Pooling for Emax parameter. Can take "rel", "common", "random" or be assigned a numeric value (see details).
onset	Pooling for onset parameter. Can take "rel", "common", "random" or be assigned a numeric value (see details).
p.expon	A logical object to indicate whether ed50 and hill parameters should be expressed within the dose-response function on an exponential scale

Details

1-parameter model: $emax \times (1 - exp(-x))$

2-parameter model: $emax \times (1 - exp(onset * -x))$

where emax is the maximum efficacy of an agent and rate is the speed

Dose-response parameter arguments:

Argument	Model specification
"rel"	Implies that *relative* effects should be pooled for this dose-response parameter separately for each agent in the n
"common"	Implies that all agents share the same common effect for this dose-response parameter.
"random"	Implies that all agents share a similar (exchangeable) effect for this dose-response parameter. This approach allo
numeric()	Assigned a numeric value, indicating that this dose-response parameter should not be estimated from the data bu

Value

An object of class("dosefun")

References

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

Examples

```
# Single parameter exponential function is default
dexp()

# Two parameter exponential function
dexp(onset="rel")
```

---

dfpoly *Fractional polynomial dose-response function*

Description

Fractional polynomial dose-response function

Usage

```
dfpoly(degree = 1, beta.1 = "rel", beta.2 = "rel", power.1 = 0, power.2 = 0)
```

Arguments

degree	The degree of the fractional polynomial as defined in Royston and Altman (1994)
beta.1	Pooling for the 1st fractional polynomial coefficient. Can take "rel", "common", "random" or be assigned a numeric value (see details).
beta.2	Pooling for the 2nd fractional polynomial coefficient. Can take "rel", "common", "random" or be assigned a numeric value (see details).
power.1	Value for the 1st fractional polynomial power (γ_1). Must take any numeric value in the set -2, -1, -0.5, 0, 0.5, 1, 2, 3.
power.2	Value for the 2nd fractional polynomial power (γ_2). Must take any numeric value in the set -2, -1, -0.5, 0, 0.5, 1, 2, 3.

Details

- β_1 represents the 1st coefficient.
- β_2 represents the 2nd coefficient.
- γ_1 represents the 1st fractional polynomial power

- γ_2 represents the 2nd fractional polynomial power

For a polynomial of `degree=1`:

$$\beta_1 x^{\gamma_1}$$

For a polynomial of `degree=2`:

$$\beta_1 x^{\gamma_1} + \beta_2 x^{\gamma_2}$$

x^γ is a regular power except where $\gamma = 0$, where $x^{(0)} = ln(x)$. If a fractional polynomial power γ repeats within the function it is multiplied by another $ln(x)$.

Value

An object of `class("dosefun")`

Dose-response parameters

Argument	Model specification
`"rel"`	Implies that *relative* effects should be pooled for this dose-response parameter separately for each agent in the n
`"common"`	Implies that all agents share the same common effect for this dose-response parameter.
`"random"`	Implies that all agents share a similar (exchangeable) effect for this dose-response parameter. This approach allo
`numeric()`	Assigned a numeric value, indicating that this dose-response parameter should not be estimated from the data bu

When relative effects are modelled on more than one dose-response parameter, correlation between them is automatically estimated using a vague inverse-Wishart prior. This prior can be made slightly more informative by specifying the scale matrix `omega` and by changing the degrees of freedom of the inverse-Wishart prior using the `priors` argument in `mbnma.run()`.

References

Royston P, Altman D (1994). "Regression Using Fractional Polynomials of Continuous Covariates: Parsimonious Parametric Modelling." *Journal of the Royal Statistical Society: Series C*, **43**(3), 429-467.

Examples

```
# 1st order fractional polynomial a value of 0.5 for the power
dfpoly(beta.1="rel", power.1=0.5)

# 2nd order fractional polynomial with relative effects for coefficients
# and a value of -0.5 and 2 for the 1st and 2nd powers respectively
dfpoly(degree=2, beta.1="rel", beta.2="rel",
  power.1=-0.5, power.2=2)
```

`ditp` *Integrated Two-Component Prediction (ITP) function*

Description

Similar parameterisation to the Emax model but with non-asymptotic maximal effect (Emax). Proposed by proposed by Fu and Manner (2010)

Usage

```
ditp(emax = "rel", rate = "rel", p.expon = FALSE)
```

Arguments

`emax`	Pooling for Emax parameter. Can take `"rel"`, `"common"`, `"random"` or be assigned a numeric value (see details).
`rate`	Pooling for Rate parameter. Can take `"rel"`, `"common"`, `"random"` or be assigned a numeric value (see details).
`p.expon`	A logical object to indicate whether `ed50` and `hill` parameters should be expressed within the dose-response function on an exponential scale

Details

Emax represents the maximum response. Rate represents the rate at which a change in the dose of the drug leads to a change in the effect

$$E_{max} \times \frac{(1 - exp(-rate \times x))}{(1 - exp(-rate \times max(x)))}$$

Value

An object of `class("dosefun")`

Dose-response parameters

Argument	Model specification
`"rel"`	Implies that *relative* effects should be pooled for this dose-response parameter separately for each agent in the n
`"common"`	Implies that all agents share the same common effect for this dose-response parameter.
`"random"`	Implies that all agents share a similar (exchangeable) effect for this dose-response parameter. This approach allo
`numeric()`	Assigned a numeric value, indicating that this dose-response parameter should not be estimated from the data bu

When relative effects are modelled on more than one dose-response parameter, correlation between them is automatically estimated using a vague inverse-Wishart prior. This prior can be made slightly more informative by specifying the scale matrix `omega` and by changing the degrees of freedom of the inverse-Wishart prior using the `priors` argument in `mbnma.run()`.

References

Fu H, Manner D (2010). "Bayesian adaptive dose-finding studies with delayed responses." *J Biopharm Stat*, **20**(5), 1055-1070. doi:10.1080/10543400903315740.

Examples

```
# Model a common effect on rate
ditp(emax="rel", rate="common")
```

`dloglin` *Log-linear (exponential) dose-response function*

Description

Modelled assuming relative effects (`"rel"`)

Usage

```
dloglin()
```

Details

$rate \times log(x + 1)$

Dose-response parameter arguments:

Argument	Model specification
`"rel"`	Implies that *relative* effects should be pooled for this dose-response parameter separately for each agent in the n
`"common"`	Implies that all agents share the same common effect for this dose-response parameter.
`"random"`	Implies that all agents share a similar (exchangeable) effect for this dose-response parameter. This approach allo
`numeric()`	Assigned a numeric value, indicating that this dose-response parameter should not be estimated from the data bu

Value

An object of `class("dosefun")`

References

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

Examples

```
dloglin()
```

dmulti *Agent-specific dose-response function*

Description

Function combines different dose-response functions together to create an object containing parameters for multiple dose-response functions.

Usage

```
dmulti(funs = list())
```

Arguments

funs	A list of objects of `class("dosefun")`, each element of which corresponds to an agent in the dataset to be modelled. The list length must be equal to the number of agents in `network$agents` used in `mbnma.run()`, and the order of the dose-response functions in the list is assumed to correspond to the same order of agents in `network$agents`.

Value

An object of `class("dosefun")`

Examples

```
funs <- c(rep(list(demax()),3),
          rep(list(dloglin()),2),
          rep(list(demax(ed50="common")),3),
          rep(list(dexp()),2))

dmulti(funs)
```

dnonparam *Non-parameteric dose-response functions*

Description

Used to fit monotonically increasing non-parametric dose-response relationship following the method of Owen et al. (2015))

Usage

```
dnonparam(direction = "increasing")
```

Arguments

direction	Can take either "increasing" or "decreasing" to indicate the monotonic direction of the dose-response relationship

Value

An object of class("dosefun")

References

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/.

Examples

```
# Monotonically increasing dose-response
dnonparam(direction="increasing")

# Monotonically decreasing dose-response
dnonparam(direction="decreasing")
```

---

dpoly *Polynomial dose-response function*

Description

Polynomial dose-response function

Usage

```
dpoly(
  degree = 1,
  beta.1 = "rel",
  beta.2 = "rel",
  beta.3 = "rel",
  beta.4 = "rel"
)
```

Arguments

degree	The degree of the polynomial - e.g. degree=1 for linear, degree=2 for quadratic, degree=3 for cubic.
beta.1	Pooling for the 1st polynomial coefficient. Can take "rel", "common", "random" or be assigned a numeric value (see details).

`beta.2`	Pooling for the 2nd polynomial coefficient. Can take `"rel"`, `"common"`, `"random"` or be assigned a numeric value (see details).
`beta.3`	Pooling for the 3rd polynomial coefficient. Can take `"rel"`, `"common"`, `"random"` or be assigned a numeric value (see details).
`beta.4`	Pooling for the 4th polynomial coefficient. Can take `"rel"`, `"common"`, `"random"` or be assigned a numeric value (see details).

Details

- β_1 represents the 1st coefficient.
- β_2 represents the 2nd coefficient.
- β_3 represents the 3rd coefficient.
- β_4 represents the 4th coefficient.

Linear model:

$$\beta_1 x$$

Quadratic model:

$$\beta_1 x + \beta_2 x^2$$

Cubic model:

$$\beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

Quartic model:

$$\beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4$$

Value

An object of `class("dosefun")`

Dose-response parameters

Argument	Model specification
`"rel"`	Implies that *relative* effects should be pooled for this dose-response parameter separately for each agent in the n
`"common"`	Implies that all agents share the same common effect for this dose-response parameter.
`"random"`	Implies that all agents share a similar (exchangeable) effect for this dose-response parameter. This approach allo
`numeric()`	Assigned a numeric value, indicating that this dose-response parameter should not be estimated from the data bu

When relative effects are modelled on more than one dose-response parameter, correlation between them is automatically estimated using a vague inverse-Wishart prior. This prior can be made slightly more informative by specifying the scale matrix `omega` and by changing the degrees of freedom of the inverse-Wishart prior using the `priors` argument in `mbnma.run()`.

References

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

Examples

```
# Linear model with random effects
dpoly(beta.1="rel")

# Quadratic model dose-response function
# with an exchangeable (random) absolute parameter estimated for the 2nd coefficient
dpoly(beta.1="rel", beta.2="random")
```

`DR.comparisons` *Adds placebo comparisons for dose-response relationship*

Description

Function adds additional rows to a data.frame of comparisons in a network that account for the relationship between placebo and other agents via the dose-response relationship.

Usage

```
DR.comparisons(data.ab, level = "treatment", doselink = NULL)
```

Arguments

`data.ab`	A data frame stored in an `mbnma.network` object (`mbnma.network$data.ab`)
`level`	A character that can take either `"treatment"` or `"agent"` to indicate the level of the network for which to identify dose-response
`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.

`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(triptans)
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))
```

`dspline` *Spline dose-response functions*

Description

Used to fit B-splines and natural cubic splines(Perperoglu et al. 2019).

Usage

```
dspline(type = "bs", knots = NULL, degree = 3, df = NULL, betas = "rel")
```

Arguments

`type`	The type of spline. Can take `"bs"` (B-spline), or `"ns"` (natural cubic spline). Piecewise linear splines can be fitted using `"bs"` with `degree=1`.
`knots`	Indicates the number/location of internal knots. If a single whole number `>=1` is given it indicates the number of equally-spaced internal knots. Otherwise (a vector, or a non-integer value) the values are treated as the quantile locations of the knots as a proportion of the maximum dose in the dataset. For example, if the maximum dose in the dataset is 100mg/d for a particular agent, `knots=c(0.1,0.5)` would indicate knots should be fitted at 10mg/d and 50mg/d.
`degree`	a positive integer giving the degree of the polynomial from which the spline function is composed (e.g. `degree=3` represents a cubic spline).
`df`	degrees of freedom. One can supply `df` rather than knots; `ns()` then chooses `df - 1 - intercept` knots at suitably chosen quantiles of `x` (which will ignore missing values). The default, `df = NULL`, sets the number of inner knots as `length(knots)`.
`betas`	A vector of beta parameters corresponding to each spline coefficient. The length must be equal to the number of dose-response parameters generated by the spline function. If a single value is given then this specification will be applied to all beta paramters in the model. Can take `"rel"`, `"common"`, `"random"` or be assigned a numeric value (see details).

Value

An object of `class("dosefun")`

Dose-response parameters

Argument	Model specification
"rel"	Implies that *relative* effects should be pooled for this dose-response parameter separately for each agent in the n
"common"	Implies that all agents share the same common effect for this dose-response parameter.
"random"	Implies that all agents share a similar (exchangeable) effect for this dose-response parameter. This approach allo
numeric()	Assigned a numeric value, indicating that this dose-response parameter should not be estimated from the data bu

When relative effects are modelled on more than one dose-response parameter, correlation between them is automatically estimated using a vague inverse-Wishart prior. This prior can be made slightly more informative by specifying the scale matrix `omega` and by changing the degrees of freedom of the inverse-Wishart prior using the `priors` argument in `mbnma.run()`.

References

Perperoglu A, Sauerbrei W, Abrahamowicz M, Schmid M (2019). "A review of spline function procedures in R." *BMC Medical Research Methodology*, **19**(46), 1-16. doi:10.1186/s12874019-06663.

Examples

```
# Natural cubic spline with 1 knot
dspline(type="bs", knots=1,
  betas="rel")

# Piecewise linear B-spline with knots at 0.1 and 0.5 quantiles
# Single parameter independent of treatment estimated for 1st coefficient
#with random effects
dspline(type="bs", degree=1, knots=c(0.1,0.5),
  betas=c("random", "rel", "rel"))
```

`duser` *User-defined dose-response function*

Description

User-defined dose-response function

Usage

```
duser(fun, beta.1 = "rel", beta.2 = "rel", beta.3 = "rel", beta.4 = "rel")
```

Arguments

fun	A formula specifying any relationship including dose and one/several of: beta.1, beta.2, beta.3, beta.4.
beta.1	Pooling for the 1st coefficient. Can take "rel", "common", "random" or be assigned a numeric value (see details).
beta.2	Pooling for the 2nd coefficient. Can take "rel", "common", "random" or be assigned a numeric value (see details).
beta.3	Pooling for the 3rd coefficient. Can take "rel", "common", "random" or be assigned a numeric value (see details).
beta.4	Pooling for the 4th coefficient. Can take "rel", "common", "random" or be assigned a numeric value (see details).

Value

An object of class("dosefun")

Dose-response parameters

Argument	Model specification
"rel"	Implies that *relative* effects should be pooled for this dose-response parameter separately for each agent in the n
"common"	Implies that all agents share the same common effect for this dose-response parameter.
"random"	Implies that all agents share a similar (exchangeable) effect for this dose-response parameter. This approach allo
numeric()	Assigned a numeric value, indicating that this dose-response parameter should not be estimated from the data bu

When relative effects are modelled on more than one dose-response parameter, correlation between them is automatically estimated using a vague inverse-Wishart prior. This prior can be made slightly more informative by specifying the scale matrix omega and by changing the degrees of freedom of the inverse-Wishart prior using the priors argument in mbnma.run().

References

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

Examples

```
dr <- ~ beta.1 * (1/(dose+1)) + beta.2 * dose^2
duser(fun=dr,
  beta.1="common", beta.2="rel")
```

`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(triptans)

# Run an Emax dose-response MBNMA and predict responses
emax <- mbnma.run(network, fun=demax(), 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)
```

---

gen.parameters.to.save

Automatically generate parameters to save for a dose-response MBNMA model

---

Description

Automatically generate parameters to save for a dose-response MBNMA model

Usage

```
gen.parameters.to.save(fun, model, regress.mat = NULL)
```

Arguments

fun	An object of `class("dosefun")` that specifies a functional form to be assigned to the dose-response. See Details.
model	A JAGS model written as a character object
regress.mat	A Nstudy x Ncovariate design matrix of meta-regression covariates

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 = "bs",
  df = 1,
  knots = NULL,
  degree = 3,
  max.dose = max(x),
  boundaries = NULL
)
```

Arguments

x	A numeric vector indicating all time points available in the dataset
spline	Indicates the type of spline function. Can be either a natural cubic spline ("ns"), or B-spline ("bs").
df	degrees of freedom. One can supply df rather than knots; ns() then chooses df - 1 - intercept knots at suitably chosen quantiles of x (which will ignore missing values). The default, df = NULL, sets the number of inner knots as length(knots).
knots	Indicates the number/location of internal knots. If a single whole number >=1 is given it indicates the number of equally-spaced internal knots. Otherwise (a vector, or a non-integer value) the values are treated as the quantile locations of the knots as a proportion of the maximum dose in the dataset. For example, if the maximum dose in the dataset is 100mg/d for a particular agent, knots=c(0.1,0.5) would indicate knots should be fitted at 10mg/d and 50mg/d.
degree	a positive integer giving the degree of the polynomial from which the spline function is composed (e.g. degree=3 represents a cubic spline).
max.dose	A number indicating the maximum dose between which to calculate the spline function.
boundaries	A positive numeric vector of length 2 that represents the doses at which to anchor the B-spline or natural cubic spline basis matrix. This allows data to extend beyond the boundary knots, or for the basis parameters to not depend on x. The default (boundaries=NULL) is the range of x.

Value

A spline basis matrix with number of rows equal to length(x) and the number of columns equal to the number of coefficients in the spline.

Examples

```
x <- 0:100

genspline(x)

# Generate a quadratic B-spline with 1 equally spaced internal knot
genspline(x, spline="bs", df=2, degree=2)

# Generate a natural cubic spline with 3 knots at selected quantiles
genspline(x, spline="ns", knots=c(0.1, 0.5, 0.7))

# Generate a piecewise linear spline with 2 equally spaced knots
genspline(x, spline="bs", degree=1, df=3)
```

---

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(triptans)

# Run an Emax dose-response MBNMA
result <- mbnma.run(network, fun=demax(), 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=dexp(), method="random")
print(result$model.arg$priors)
```

`get.relative` *Calculates league table of effects between treatments in MBNMA and/or NMA models*

Description

Calculates league table of effects between treatments in MBNMA and/or NMA models

Usage

```
get.relative(
  lower.diag,
  upper.diag = lower.diag,
  treatments = list(),
  lower.direction = "colvrow",
  upper.direction = "rowvcol",
  regress.vals = NULL,
  eform = FALSE,
  lim = "cred"
)
```

Arguments

`lower.diag`	An S3 object either of class `"mbnma"` generated by running a dose-response MBNMA model (using `mbnma.run()`), or of class `"nma"` generated by running an NMA model (using `nma.run()`). Indicates that this model should be used to estimate treatment effects in the **lower-left diagonal** of the league table.
`upper.diag`	Same as for `lower.diag`, but indicates that this model should be used to estimate treatment effects in the **upper-right diagonal** of the league table.

`treatments` A list whose elements each represent different treatments. Treatment is defined as a combination of agent and dose. Only agents specified in `lower.diag` and `upper.diag` 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`. If `treatments` is left empty then the set of treatments in the network on which `lower.diag` is estimated will be used as the default.

`lower.direction` Whether treatment effects should be presented as the column versus the row treatment for each cell in the **lower-left diagonal** of the league table (`"colvrow"`) or as the row versus the column treatment (`"rowvcol"`).

`upper.direction` Same as for `upper.direction` but for results shown in the **upper-right diagonal** of the league table.

`regress.vals` A named numeric vector of effect modifier values at which relative effects should be estimated. Named elements must match variable names specified in regression design matrix (`mbnma$model.arg$regress.mat`), corresponding to variables in `regress` within the MBNMA model (or **both** MBNMA models if different models are used for `upper.diag` and `lower.diag`. Meta-regression is not currently possible for NMA models, so only unadjusted results will be shown for these (and a warning given if they are compared to adjusted MBNMA results).

`eform` Whether outputted results should be presented in their exponential form (e.g. for models with log or logit link functions)

`lim` Specifies calculation of either 95% credible intervals (`lim="cred"`) or 95% prediction intervals (`lim="pred"`).

Value

An array of `length(treatments) x length(treatments) x nsims`, where `nsims` is the number of iterations monitored in `lower.diag`. The array contains the individual MCMC values for each relative effect calculated between all `treatments` on the link scale specified in the `lower.diag` and `upper.diag` models.

Examples

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

# Run an MBNMA model
expon <- mbnma.run(network, fun=dexp(), method="random")

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

# Run an NMA model
nma <- nma.run(network, method="random")
```

```
# Compare results between MBNMA and NMA models
rel.eff <- get.relative(lower.diag=expon, upper.diag=nma,
  treatments=list("Celebrex"=c(100,200), "Tramadol"=100),
  upper.direction="colvrow")
```

`getjagsdata` *Prepares data for JAGS*

Description

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

Usage

```
getjagsdata(
  data.ab,
  class = FALSE,
  sdscale = FALSE,
  regress = NULL,
  regress.effect = "common",
  likelihood = check.likelink(data.ab)$likelihood,
  link = check.likelink(data.ab)$link,
  level = "agent",
  fun = NULL,
  nodesplit = 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 or when modelling Standardised Mean Differences
- `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.
- `standsd` An optional column of numeric data indicating reference SDs used to standardise treatment effects when modelling using Standardised Mean Differences (SMD).

`class` A boolean object indicating whether or not `data.ab` contains information on different classes of treatments

`sdscale` Logical object to indicate whether to write a model that specifies a reference SD for standardising when modelling using Standardised Mean Differences. Specifying `sdscale=TRUE` will therefore only modify the model if link function is set to SMD (`link="smd"`).

`regress` A formula of effect modifiers (variables that interact with the treatment effect) to incorporate using Network Meta-Regression (E.g. `~ Population + Age`). All variables in the formula are modelled as interacting with the treatment effect (i.e. prognostic variables cannot be included in this way). Effects modifiers must be named variables in `network$data.ab` and must be identical within a study. Factor and character effect modifiers will be converted to a series of named dummy variables.

`regress.effect` Indicates whether effect modification should be assumed to be `"common"` (assumed to be equal versus Placebo throughout the network), `"random"` (assumed to be exchangeable versus Placebo throughout the network), `"agent"` (assumed to be equal versus Placebo within each agent), or `"class"` (assumed to be equal versus Placebo within each class).

`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"`), be assigned the value `"identity"` for an identity link function, or be assigned the value `"smd"` for modelling Standardised Mean Differences using an 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` An object of `class("dosefun")` that specifies a functional form to be assigned to the dose-response. See Details.

`nodesplit` A numeric vector of length 2 containing treatment codes on which to perform an MBNMA nodesplit (see `mbnma.nodesplit`).

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


# Get JAGS data with class
netclass <- mbnma.network(osteopain)
jagsdat <- getjagsdata(netclass$data.ab, class=TRUE)


# Get JAGS data at the treatment level for split Network Meta-Analysis
network <- mbnma.network(triptans)
jagsdat <- getjagsdata(network$data.ab, level="treatment")
```

gout — *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

```
gout
```

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.

inconsistency.loops — *Identify comparisons in loops that fulfill criteria for node-splitting*

Description

Identify comparisons informed by both direct and indirect evidence from independent sources, which therefore fulfill the criteria for testing for inconsistency via node-splitting.

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

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

Examples

```
# Identify comparisons informed by direct and indirect evidence
#in triptans dataset
network <- mbnma.network(triptans)
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(df)
```

Arguments

`df`	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

```
df <- 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(df)


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

`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 = dpoly(degree = 1),
  method = "common",
  comparisons = NULL,
  incldr = TRUE,
  ...
)

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

Arguments

`network`	An object of class `mbnma.network`.
`fun`	An object of `class("dosefun")` that specifies a functional form to be assigned to the dose-response. 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).
`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()` or `ggplot2::geom_pointrange()`
`x`	An object of `class("nodesplit")`
`plot.type`	A character string that can take the value of `"forest"` to plot forest plots or `"density"` to plot posterior density plots.

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 and invisibly returns an object of `class(c("gg", "ggplot"))` for both `plot.type="forest"` and `plot.type="density"`.

Functions

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

Examples

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

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

#...is the same as...
single.split <- mbnma.nodesplit(network, fun=dexp(), 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

# 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 = dpoly(degree = 1),
  method = "common",
  regress = NULL,
  regress.effect = "common",
  class.effect = list(),
  UME = FALSE,
  sdscale = FALSE,
  cor = FALSE,
  omega = NULL,
  parameters.to.save = NULL,
  pD = TRUE,
  likelihood = NULL,
  link = NULL,
  priors = NULL,
  n.iter = 20000,
  n.chains = 3,
  n.burnin = floor(n.iter/2),
  n.thin = max(1, floor((n.iter - n.burnin)/1000)),
  autojags = FALSE,
  Rhat = 1.05,
  n.update = 10,
  model.file = NULL,
  jagsdata = NULL,
  ...
)
```

Arguments

`network`	An object of class `mbnma.network`.
`fun`	An object of `class("dosefun")` that specifies a functional form to be assigned to the dose-response. 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).

regress — A formula of effect modifiers (variables that interact with the treatment effect) to incorporate using Network Meta-Regression (E.g. ~ Population + Age). All variables in the formula are modelled as interacting with the treatment effect (i.e. prognostic variables cannot be included in this way). Effects modifiers must be named variables in network$data.ab and must be identical within a study. Factor and character effect modifiers will be converted to a series of named dummy variables.

regress.effect — Indicates whether effect modification should be assumed to be "common" (assumed to be equal versus Placebo throughout the network), "random" (assumed to be exchangeable versus Placebo throughout the network), "agent" (assumed to be equal versus Placebo within each agent), or "class" (assumed to be equal versus Placebo within each class).

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. Note that assuming class effects on some dose-response parameters may be unreasonable if the range of doses differ substantially across agents within a class.

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.

sdscale — Logical object to indicate whether to write a model that specifies a reference SD for standardising when modelling using Standardised Mean Differences. Specifying sdscale=TRUE will therefore only modify the model if link function is set to SMD (link="smd").

cor — A boolean object that indicates whether correlation should be modelled between relative effect dose-response parameters. This is automatically set to FALSE if class effects are modelled or if multiple dose-response functions are fitted.

omega — A scale matrix for the inverse-Wishart prior for the covariance matrix used to model the correlation between dose-response parameters (see Details for dose-response functions). omega must be a symmetric positive definite matrix with dimensions equal to the number of dose-response parameters modelled using relative effects ("rel"). If left as NULL (the default) a diagonal matrix with elements equal to 100 is used.

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

pD — logical; if TRUE (the default) then adds the computation of pD, using the method of (Plummer 2008). If FALSE then uses the approximation of pD=var(deviance) / 2 (often referred to as pV).

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"), be assigned the value "identity" for an identity link function, or be assigned the value "smd" for modelling Standardised Mean Differences using an 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 Plummer (2017)). Note that normal distributions in JAGS are specified as

$$N(\mu, prec)$$

, where

$$prec = 1/\sigma^2$$

.

n.iter — number of total iterations per chain (including burn in; default: 20000)

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 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 `1.05`.

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).

model.file — The file path to a JAGS model (.jags file) that can be used to overwrite the JAGS model that is automatically written based on the specified options in `MBNMAdose`. Useful for adding further model flexibility.

jagsdata — A named list of the data objects to be used in the JAGS model. Only required if users are defining their own JAGS model using `model.file`. Format should match that of standard models fitted in `MBNMAdose` (see `mbnma$model.arg$jagsdata`)

... — 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 `omega`. `cor` will automatically be set to `FALSE` if class effects are modelled.

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 `coda`.

Nodes that are automatically monitored (if present in the model) have the following interpretation:

Parameters(s)/Parameter Prefix	Interpretation
<named dose-response parameter> (e.g. emax)	The pooled effect for each dose-response paramete
sd	The between-study SD (heterogeneity) for relative
sd.<named dose-response parameter> (e.g. sd.emax)	Between-study SD (heterogeneity) for absolute do
<named capitalized dose-response parameter> (e.g. EMAX)	The class effect within each class for a given dose-
sd.<named capitalized dose-response parameter> (e.g. sd.EMAX)	The within-class SD for different agents within the
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 parameter arguments

Argument	Model specification
"rel"	Implies that *relative* effects should be pooled for this dose-response parameter separately for each agent in the n
"common"	Implies that all agents share the same common effect for this dose-response parameter.
"random"	Implies that all agents share a similar (exchangeable) effect for this dose-response parameter. This approach allo
numeric()	Assigned a numeric value, indicating that this dose-response parameter should not be estimated from the data bu

Dose-response function

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

As of version 0.4.0 dose-response functions are specified as an object of `class("dosefun")`. See help details for each of the functions below for the interpretation of specific dose-response parameters.

Built-in dose-response functions are:

- `dpoly()`: polynomial (e.g. for a linear model - `dpoly(degree=1)`)
- `dloglin()`: log-linear
- `dexp()`: exponential
- `demax()`: (emax with/without a Hill parameter)
- `dspline()`: splines (can fit B-splines (`type="bs"`) or natural splines (`type="ns"`) of any degree
- `dfpoly()`: fractional polynomials
- `dnonparam()`: Non-parametric monotonic function (`direction` can be either `"increasing"` or `"decreasing"`) following the method of Owen et al. (2015)

- `duser()`: user-defined function
- `dmulti()`: allows agent-specific dose-response functions to be fitted. A separate function must be provided for each agent in the network.

References

Gelman A, Rubin DB (1992). "Inference from iterative simulation using multiple sequences." *Statistical Science*, **7**(4), 457-511. https://projecteuclid.org/journals/statistical-science/volume-7/issue-4/Inference-from-Iterative-Simulation-Using-Multiple-Sequences/10.1214/ss/1177011136.full.

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/.

Plummer M (2017). *JAGS user manual*. https://people.stat.sc.edu/hansont/stat740/jags_user_manual.pdf.

Examples

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


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

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

# Fit a dose-response MBNMA with a log-linear function
# with random treatment effects
result <- mbnma.run(network, fun=dloglin(), method="random")

# Fit a dose-response MBNMA with a fractional polynomial function
# with random treatment effects
# with a probit link function
result <- mbnma.run(network, fun=dfpoly(), method="random", link="probit")

# Fit a user-defined function (quadratic)
```

```
fun.def <- ~ (beta.1 * dose) + (beta.2 * (dose^2))
result <- mbnma.run(network, fun=duser(fun=fun.def), method="common")

# Fit an Emax function
# with a single random (exchangeable) parameter for ED50
# with common treatment effects
result <- mbnma.run(network, fun=demax(emax="rel", ed50="random"),
              method="common")

# Fit an Emax function with a Hill parameter
# with a fixed value of 5 for the Hill parameter
# with random relative effects
result <- mbnma.run(network, fun=demax(hill=5), method="random")

# Fit a model with natural cubic splines
# with 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=dspline(type="ns", knots=c(0.1,0.3,0.6)))

# Fit a model with different dose-response functions for each agent
multifun <- dmulti(list(dloglin(), # for placebo (can be any function)
                        demax(), # for eletriptan
                        demax(), # for sumatriptan
                        dloglin(), # for frovatriptan
                        demax(), # for almotriptan
                        demax(), # for zolmitriptan
                        dloglin(), # for naratriptan
                        demax())) # for rizatriptan

multidose <- mbnma.run(network, fun=multifun)


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

 # Using the osteoarthritis dataset
 pain.df <- osteopain

 # Set a shared class (NSAID) only for Naproxcinod and Naproxen
 pain.df <- pain.df %>% dplyr::mutate(
              class = dplyr::case_when(agent %in% c("Naproxcinod", "Naproxen") ~ "NSAID",
                        !agent %in% c("Naproxcinod", "Naproxen") ~ agent
                        )
                )

 # Run an Emax MBNMA with a common class effect on emax
 painnet <- mbnma.network(pain.df)
 result <- mbnma.run(painnet, fun = demax(),
                class.effect = list(emax = "common"))


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

# Obtain priors from a fractional polynomial function
```

```
result <- mbnma.run(network, fun=dfpoly(degree=1), method="random")
print(result$model.arg$priors)

# Change the prior distribution for the power
newpriors <- list(power.1 = "dnorm(0,0.001) T(0,)")
newpriors <- list(sd = "dnorm(0,0.5) T(0,)")

result <- mbnma.run(network, fun=dfpoly(degree=1), method="random",
              priors=newpriors)


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

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

####### Examine MCMC diagnostics (using the coda package) #######

if (requireNamespace("coda")) {
  # Density plots
  coda::densplot(coda::as.mcmc(result))

  # Traceplots
  coda::traceplot(coda::as.mcmc(result))

  # Autocorrelation plots
  coda::autocorr.plot(coda::as.mcmc(result))
}

####### 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=demax(),
              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. Same function used in MBNMAdose and MBNMAtime packages.

Usage

```
mbnma.update(
  mbnma,
  param = "theta",
  armdat = 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
`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
`armdat`	Include raw arm-level data for each data point (agent, dose, study grouping)
`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(triptans)

# Fit a dose-response MBNMA, monitoring "psi" and "resdev"
result <- mbnma.run(network, fun=dloglin(), 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 fulfills requirements for MBNMA*

Description

Validates that a dataset fulfills 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 or when modelling Standardised Mean Differences
- `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.
- `standsd` An optional column of numeric data indicating reference SDs used to standardise treatment effects when modelling using Standardised Mean Differences (SMD).

`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
- Checks that agent names do not include underscores
- Checks that standsd values are consistent within a study

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 = dpoly(degree = 1),
  method = "common",
  regress.mat = NULL,
  regress.effect = "common",
  sdscale = FALSE,
  cor = FALSE,
  cor.prior = "wishart",
  omega = NULL,
  om = list(rel = 5, abs = 10),
  class.effect = list(),
  UME = FALSE,
  likelihood = "binomial",
  link = NULL
)
```

Arguments

`fun`	An object of `class("dosefun")` that specifies a functional form to be assigned to the dose-response. 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).
`regress.mat`	A Nstudy x Ncovariate design matrix of meta-regression covariates
`regress.effect`	Indicates whether effect modification should be assumed to be `"common"` (assumed to be equal versus Placebo throughout the network), `"random"` (assumed to be exchangeable versus Placebo throughout the network), `"agent"` (assumed to be equal versus Placebo within each agent), or `"class"` (assumed to be equal versus Placebo within each class).
`sdscale`	Logical object to indicate whether to write a model that specifies a reference SD for standardising when modelling using Standardised Mean Differences. Specifying `sdscale=TRUE` will therefore only modify the model if link function is set to SMD (`link="smd"`).
`cor`	A boolean object that indicates whether correlation should be modelled between relative effect dose-response parameters. 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"`
`omega`	A scale matrix for the inverse-Wishart prior for the covariance matrix used to model the correlation between dose-response parameters (see Details for dose-response functions). `omega` must be a symmetric positive definite matrix with dimensions equal to the number of dose-response parameters modelled using relative effects (`"rel"`). If left as `NULL` (the default) a diagonal matrix with elements equal to 100 is used.
`om`	a list with two elements that report the maximum relative (`"rel"`) and maximum absolute (`"abs"`) efficacies on the link scale.

`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. Note that assuming class effects on some dose-response parameters may be unreasonable if the range of doses differ substantially across agents within a class.

`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.

`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"`), be assigned the value `"identity"` for an identity link function, or be assigned the value `"smd"` for modelling Standardised Mean Differences using an 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 `omega`. `cor` will automatically be set to `FALSE` if class effects are modelled.

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,
# with random treatment effects
model <- mbnma.write(fun=dexp(),
             method="random",
             likelihood="binomial",
             link="logit"
             )
names(model) <- NULL
print(model)

# Write model code for a model with an Emax dose-response function,
# relative effects modelled on Emax with a random effects model,
# a single parameter estimated for ED50 with a common effects model
model <- mbnma.write(fun=demax(emax="rel", ed50="common"),
             likelihood="normal",
             link="identity"
             )
```

```
names(model) <- NULL
print(model)

# Write model code for a model with an Emax dose-response function,
# relative effects modelled on Emax and ED50.
# Class effects modelled on ED50 with common effects
model <- mbnma.write(fun=demax(),
             likelihood="normal",
             link="identity",
             class.effect=list("ed50"="common")
             )
names(model) <- NULL
print(model)

# Write model code for a model with an Emax dose-response function,
# relative effects modelled on Emax and ED50 with a
# random effects model that automatically models a correlation between
# both parameters.
model <- mbnma.write(fun=demax(),
             method="random",
             likelihood="normal",
             link="identity",
             )
names(model) <- NULL
print(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 subsequently MBNMA) 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"), be assigned the value "identity" for an identity link function, or be assigned the value "smd" for modelling Standardised Mean Differences using an 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(nma.nodesplit): Plot outputs from treatment-level nodesplit models

Examples

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

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
```

`norm2lnorm` *Convert normal distribution parameters to corresponding log-normal distribution parameters*

Description

Converts mean and variance of normal distribution to the parameters for a log-normal distribution with the same mean and variance

Usage

```
norm2lnorm(m, v)
```

Arguments

`m`	Mean of the normal distribution
`v`	Variance of the normal distribution

Value

A vector of length two. The first element is the mean and the second element is the variance of the log-normal distribution

Examples

```
norm <- rnorm(1000, mean=5, sd=2)
params <- norm2lnorm(5, 2^2)
lnorm <- rlnorm(1000, meanlog=params[1], sdlog=params[2]^0.5)

# Mean and SD of lnorm is equivalent to mean and sd of norm
mean(lnorm)
sd(lnorm)
```

---

osteopain *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
```

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
- `n` Numeric data indicating the number of participants randomised

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. Currently only functions with univariate likelihoods. Function is identical in MBNMAdose and MBNMAtime packages.

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(triptans)

# Fit a dose-response MBNMA, monitoring "psi" and "resdev"
result <- mbnma.run(network, fun=dloglin(), 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, ...)
```

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`.
`...`	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(triptans)

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

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


#### Forest plots including class effects ####
# Generate some classes for the data
class.df <- triptans
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.run(netclass, fun=demax(), method="random",
            class.effect=list("ed50"="common"))
```

`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 or when modelling Standardised Mean Differences
- 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.
- standsd An optional column of numeric data indicating reference SDs used to standardise treatment effects when modelling using Standardised Mean Differences (SMD).

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)
- studyID A character vector with the IDs of included studies
- 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(mbnma.network): Generate a network plot

Examples

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

# 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(triptans)

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

summary(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 or R2jags::jags

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(triptans)
```

```
# Run an Emax dose-response MBNMA and predict responses
emax <- mbnma.run(network, fun=demax(), 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(triptans)

# Estimate rankings  from an Emax dose-response MBNMA
emax <- mbnma.run(network, fun=demax(), 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="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)


# Trying to plot a parameter that has not been ranked will return an error
#### ERROR ####
# plot(ranks, params="not.a.parameter")
```

---

plot.nma *Run an NMA model*

Description

Used for calculating treatment-level NMA results, either when comparing MBNMA models to models that make no assumptions regarding dose-response , 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 at the treatment-level.

Usage

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

nma.run(
  network,
  method = "common",
  likelihood = NULL,
  link = NULL,
  priors = NULL,
  sdscale = FALSE,
  warn.rhat = TRUE,
  n.iter = 20000,
  drop.discon = TRUE,
  UME = FALSE,
  pD = TRUE,
  parameters.to.save = NULL,
  ...
)
```

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"`), be assigned the value `"identity"` for an identity link function, or be assigned the value `"smd"` for modelling Standardised Mean Differences using an 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 Plummer (2017)). Note that normal distributions in JAGS are specified as

$$N(\mu, prec)$$

, where

$$prec = 1/\sigma^2$$

.

sdscale Logical object to indicate whether to write a model that specifies a reference SD for standardising when modelling using Standardised Mean Differences. Specifying sdscale=TRUE will therefore only modify the model if link function is set to SMD (link="smd").

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: 20000)

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 logical; if TRUE (the default) then adds the computation of pD, using the method of (Plummer 2008). If FALSE then uses the approximation of pD=var(deviance) / 2 (often referred to as pV).

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

Functions

- plot(nma): 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. If Placebo (or any agents with dose=0) is included in the network then this will be used as the reference treatment, but if it is not then results will be plotted versus the network reference used in the NMA object (x).

Examples

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

# Run common effects NMA keeping treatments that are disconnected in the NMA
```

```
goutnet <- mbnma.network(gout)
nma <- nma.run(goutnet, method="common", drop.discon=FALSE)

# Run an Unrelated Mean Effects (UME) inconsistency model on triptans dataset
tripnet <- mbnma.network(triptans)
ume <- nma.run(tripnet, 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 = 30,
  exact.doses = NULL,
  E0 = 0.2,
  synth = "fixed",
  lim = "cred",
  regress.vals = NULL,
  ...
)
```

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 `30`.
`exact.doses`	A list of numeric vectors. 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 within the vector for that element corresponds to a dose of the agent for which to predict responses. Doses can only take positive values. For models fitted using `dspline()` making predictions at only a very small number of doses for each agent may throw an error since it can make the spline difficult to identify.

`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 `0.2` since a default of `0` will typically lead to non-sensical predictions unless an identify link function has been used for the MBNMA model in `object`.

- `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.

`lim` Specifies calculation of either 95% credible intervals (`lim="cred"`) or 95% prediction intervals (`lim="pred"`).

`regress.vals` A named numeric vector of effect modifier values at which results should be predicted. Named elements must match variable names specified in `regress.vars` within the MBNMA model.

`...` 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:

- predicts 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
- likelihood The likelihood used in the MBNMA model object
- link The link function used in the MBNMA model object
- network The dataset in mbnma.network format
- E0 A numeric vector of value(s) used for E0 in the prediction, on the link scale.

Examples

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

# Run an Emax dose-response MBNMA
emax <- mbnma.run(network, fun=demax(), 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)

# 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

print.relative.array *Print posterior medians (95% credible intervals) for table of relative effects/mean differences between treatments/classes*

Description

Print posterior medians (95% credible intervals) for table of relative effects/mean differences between treatments/classes

Usage

```
## S3 method for class 'relative.array'
print(x, digits = 2, ...)
```

Arguments

x	An object of class "relative.array" generated by get.relative()
digits	An integer indicating the number of significant digits to be used.
...	further arguments passed to knitr::kable

`psoriasis100` *Studies of biologics for treatment of moderate-to-severe psoriasis (100% improvement)*

Description

A dataset from a systematic review of Randomised-Controlled Trials (RCTs) comparing biologics at different doses and placebo (Warren et al. 2019). The outcome is the number of patients experiencing 100% improvement on the Psoriasis Area and Severity Index (PASI) measured at 12 weeks follow-up. The dataset includes 19 Randomised-Controlled Trials (RCTs), comparing 8 different biologics at different doses with placebo.

Usage

```
psoriasis100
```

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
- `r` 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.

psoriasis75 *Studies of biologics for treatment of moderate-to-severe psoriasis (>=75% improvement)*

Description

A dataset from a systematic review of Randomised-Controlled Trials (RCTs) comparing biologics at different doses and placebo (Warren et al. 2019). The outcome is the number of patients experiencing >=75% 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

```
psoriasis75
```

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
- `r` Numeric data indicating the number of participants who achieved >=75% 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.

`psoriasis90` *Studies of biologics for treatment of moderate-to-severe psoriasis (>=90% improvement)*

Description

A dataset from a systematic review of Randomised-Controlled Trials (RCTs) comparing biologics at different doses and placebo (Warren et al. 2019). The outcome is the number of patients experiencing >=90% improvement on the Psoriasis Area and Severity Index (PASI) measured at 12 weeks follow-up. The dataset includes 24 Randomised-Controlled Trials (RCTs), comparing 9 different biologics at different doses with placebo.

Usage

```
psoriasis90
```

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
- `r` Numeric data indicating the number of participants who achieved >=90% 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,
  lower_better = TRUE,
  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.
lower_better	Indicates whether negative responses are better (`TRUE`) or positive responses are better (`FALSE`)

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(triptans)

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

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


#### Ranking by class ####
# Generate some classes for the data
class.df <- triptans
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.run(netclass, fun=demax(), method="random",
            class.effect=list("ed50"="common"))

# Rank by class, with negative responses being worse
ranks <- rank(emax, level="class", lower_better=FALSE)
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

Ranks predictions at different doses from best to worst.

Usage

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

Arguments

`x`	An object on which to apply the rank method
`lower_better`	Indicates whether negative responses are better (`TRUE`) or positive responses are better (`FALSE`)
`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(triptans)

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

# Rank selected predictions from an Emax dose-response MBNMA
emax <- mbnma.run(network, fun=demax(), 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)
```

---

rank.relative.array *Rank relative effects obtained between specific doses*

Description

Ranks "relative.table" objects generated by get.relative().

Usage

```
## S3 method for class 'relative.array'
rank(x, lower_better = TRUE, ...)
```

Arguments

x	An object on which to apply the rank method
lower_better	Indicates whether negative responses are better (TRUE) or positive responses are better (FALSE)
...	Arguments to be passed to methods

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(triptans)

# Rank selected predictions from an Emax dose-response MBNMA
emax <- mbnma.run(network, fun=demax(), method="random")
rels <- get.relative(emax)
rank <- rank(rels, lower_better=TRUE)

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

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

---

recode.agent *Assigns agent or class variables numeric identifiers*

Description

Assigns agent or class variables numeric identifiers

Usage

```
recode.agent(data.ab, level = "agent")
```

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 or when modelling Standardised Mean Differences
- `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.
- `standsd` An optional column of numeric data indicating reference SDs used to standardise treatment effects when modelling using Standardised Mean Differences (SMD).

`level` Can take either `"agent"` or `"class"`

Details

Also relabels the agent for any arms in which dose = 0 to "Placebo_0"

Value

A list containing a data frame with recoded agent/class identifiers and a character vector of original agent/class names

---

`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(triptans)

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

# Data frame for synthesis can be taken from placebo arms
ref.df <- triptans[triptans$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"), be assigned the value "identity" for an identity link function, or be assigned the value "smd" for modelling Standardised Mean Differences using an identity link function. If left as NULL the link function will be automatically assigned based on the likelihood.

Value

A rescaled numeric vector

ssi_closure — *Studies of wound closure methods to reduce Surgical Site Infections (SSI)*

Description

A dataset from an ongoing systematic review examining the efficacy of different wound closure methods to reduce surgical site infections (SSI). The outcome is binary and represents the number of patients who experienced a SSI. The dataset includes 129 RCTs comparing 16 different interventions in 6 classes. This dataset is primarily used to illustrate how `MBNMAdose` can be used to perform different types of network meta-analysis without dose-response information.

Usage

```
ssi_closure
```

Format

A data frame in long format (one row per arm and study), with 281 rows and 6 variables:

- `studyID` Study identifiers
- `Year` Year of publication
- `n` Numeric data indicating the number of participants randomised
- `r` Numeric data indicating the number of participants who achieved >50% improvement in depression symptoms
- `trt` Treatment names, given as character data
- `class` Class names, given as character data

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, digits = 4, ...)
```

Arguments

`object`	An S3 object of class `"mbnma"` generated by running a dose-response MBNMA model
`digits`	The maximum number of digits for numeric columns
`...`	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

triptans *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

```
triptans
```

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.

Index

∗ **datasets**
  alog_pcfb, 5
  gout, 35
  osteopain, 55
  psoriasis100, 74
  psoriasis75, 75
  psoriasis90, 76
  ssi_closure, 85
  ssri, 85
  triptans, 89

add_index, 4
alog_pcfb, 5

calc.edx, 6
check.network, 6
cumrank, 7

default.priors, 8
demax, 9
devdev, 10
devplot, 11
dexp, 13
dfpoly, 14
ditp, 16
dloglin, 17
dmulti, 18
dnonparam, 18
dpoly, 19
DR.comparisons, 21
drop.comp, 21
drop.disconnected, 22
dspline, 23
duser, 24

fitplot, 26

gen.parameters.to.save, 27
genspline, 28
get.prior, 29
get.relative, 30
getjagsdata, 32
ggplot2::geom_pointrange(), *38*
ggplot2::ggplot(), *38*, *53*, *66*
gout, 35

igraph, *36*, *61*
inconsistency.loops, 35

layout_, *60*

mbnma.comparisons, 37
mbnma.network *(*plot.mbnma.network*)*, 59
mbnma.nodesplit, *33*, 38
mbnma.run, 40
mbnma.update, 47
mbnma.validate.data, 48
mbnma.write, 49

nma.nodesplit, 52
nma.run *(*plot.nma*)*, 65
norm2lnorm, 54

osteopain, 55

pDcalc, 56
plot.mbnma, 58
plot.mbnma.network, 59
plot.mbnma.predict, 62
plot.mbnma.rank, 64
plot.nma, 65
plot.nma.nodesplit *(*nma.nodesplit*)*, 52
plot.nodesplit *(*mbnma.nodesplit*)*, 38
predict.mbnma, 68
print.mbnma.network, 71
print.mbnma.predict, 71
print.mbnma.rank, 72
print.nma.nodesplit, 72
print.nodesplit, 73
print.relative.array, 73
psoriasis100, 74
psoriasis75, 75

psoriasis90, 76

R2jags::jags(), *69*, *83*
rank, 77
rank.mbnma, 77
rank.mbnma.predict, 79
rank.relative.array, 80
recode.agent, 81
ref.synth, 82
ref.synth(), *69*, *83*
rescale.link, 84

ssi_closure, 85
ssri, 85
summary.mbnma, 86
summary.mbnma.network, 87
summary.mbnma.predict, 87
summary.mbnma.rank, 88
summary.nma.nodesplit, 88
summary.nodesplit, 89

triptans, 89