Package ‘meta4diag’

May 8, 2026

Type Package

Title Meta-Analysis for Diagnostic Test Studies

Version 2.1.1

Date 2021-11-30

Author Jingyi Guo <jingyi.guo@ntnu.no> and Andrea Riebler <andrea.riebler@ntnu.no>

Maintainer Jingyi Guo <jingyi.guo@ntnu.no>

Description Bayesian inference analysis for bivariate meta-analysis of diagnostic test studies using integrated nested Laplace approximation with INLA. A purpose built graphic user interface is available. The installation of R package INLA is compulsory for successful usage. The INLA package can be obtained from <https://www.r-inla.org>. We recommend the testing version, which can be downloaded by running: install.packages(``INLA'', repos=c(getOption(``repos''), INLA=``https://inla.r-inla-download.org/R/testing''), dep=TRUE).

License GPL

Depends R (>= 2.10), sp, methods, grid, shiny, shinyBS, caTools

Suggests INLA

Additional_repositories https://inla.r-inla-download.org/R/testing

Repository CRAN

Repository/R-Forge/Project meta4diag

Repository/R-Forge/Revision 29

Repository/R-Forge/DateTimeStamp 2021-11-30 09:10:28

Date/Publication 2021-11-30 10:10:02 UTC

NeedsCompilation no

Contents

`meta4diag-package` *Meta-Analysis for Diagnostic Test Studies*

Description

Bayesian inference analysis for bivariate meta-analysis of diagnostic test studies using integrated nested Laplace approximation with INLA.

Details

Package:	meta4diag
Type:	Package
Version:	2.0.7
Date:	2018-02-14
License:	GPL-2
LazyLoad:	yes

The meta4diag package provides tools to implement Bayesian bivariate meta-analyses of diagnostic test studies. meta4diag is a purpose-built front end of the R package INLA (Rue H., Martino S, and Chopin N. 2009). It allows the user a straightforward model specification and offers user-specific prior distributions. Further, the newly proposed penalized complexity prior framework (Simpson et al. 2014) is supported, which builds on prior intuitions about the behaviors of the variance and correlation parameters (Guo, J., Riebler, A. and Rue H. 2017). Accurate posterior marginal distributions for sensitivity and specificity as well as all hyperparameters, and covariates are directly obtained without Markov chain Monte Carlo sampling. Further, univariate estimates of interest, such as odds ratios, as well as the summary receiver operating characteristic (SROC) curve and other common graphics are directly available for interpretation. An interactive graphical user interface provides the user with the full functionality of the package without requiring any R programming.

Author(s)

Jingyi Guo and Andrea Riebler

References

Rue H., Martino S, and Chopin N. (2009). Approximate Bayesian Inference for Latent Gaussian Models Using Integrated Nested Laplace Approximations. Journal of the Royal Statistical Society B ***71****: 319–392. (www.r-inla.org)*

Simpson DP, Martins TG, Riebler A, Fuglstad GA, Rue H, Sorbye SH (2014) Penalised Model Component Complexity: A principled, Practical Approach to Constructing Priors. Arxiv e-prints. 1403.4630

Guo, J., Riebler, A. and Rue H. (2017) Bayesian bivariate meta-analysis of diagnostic test studies with interpretable priors. Statistics in Medicine ***36(19)****: 3039–3058.*

Guo, J. and Riebler, A. (2018) meta4diag: Bayesian Bivariate Meta-Analysis of Diagnostic Test Studies for Routine Practice. Journal of Statistical Software ***83(1)****: 1–31.*

`AUC` *A function that calculate the area under summary receiver operating chracteristic line.*

Description

Takes a `meta4diag` object produced by `meta4diag()` and calculate the area under summary receiver operating chracteristic line.

Usage

```
## S3 method for class 'meta4diag'
AUC(x, sroc.type=1, est.type="median", ...)
```

Arguments

`x`	A `meta4diag` object.
`est.type`	The estimates type used to make SROC plot. Options are "mean" and "median".
`sroc.type`	A numerical value specifying the function used to make SROC line. Options are 1, 2, 3, 4, 5. When `sroc.type=1`, the SROC line is plotted as " The regression line 1" according to Arends et al.(2008),

$$y = \mu + \rho\sqrt{\frac{\sigma_\mu^2}{\sigma_\nu^2}}(x - \nu)$$

When `sroc.type=2`, the SROC line is plotted as "The major axis method",

$$y = \frac{\sigma_\mu^2 - \sigma_\nu^2 \pm \sqrt{(\sigma_\mu^2 - \sigma_\nu^2)^2 + 4\rho^2\sigma_\mu^2\sigma_\nu^2}}{2\rho\sqrt{\sigma_\mu^2\sigma_\nu^2}}(x-\nu)+\mu$$

When `sroc.type=3`, the SROC line is plotted as "The Moses and Littenberg's regression line",

$$y = \frac{\sigma_\mu^2 + \rho\sqrt{\sigma_\mu^2\sigma_\nu^2}}{\sigma_\nu^2 + \rho\sqrt{\sigma_\mu^2\sigma_\nu^2}}(x-\nu)+\mu$$

When `sroc.type=4`, the SROC line is plotted as "The regression line 2",

$$y = \mu + \frac{1}{\rho}\sqrt{}(\frac{\sigma_\mu^2}{\sigma_\nu^2})(x-\nu)$$

When `sroc.type=5`, the SROC line is plotted as "The Rutter and Gatsonis's SROC curve",

$$y = \mu + \sqrt{\frac{\sigma_\mu^2}{\sigma_\nu^2}}(x-\nu)$$

`...` Arguments to be passes to methods.

Details

The `AUC()` returns an estimate of the area under the SROC curve. When the number of samples is given in the `makeObject()` or `meta4diag()`, the uncertainty of AUC is estimated. Otherwise only the estimates of the AUC respect to the estimated SROC curve is returned.

Value

A vector containing the "est", which indicate the estimates of AUC of the estimated SROC vurve and "mean" and "sd" (plus, possibly quantiles) of AUC if `nsample` is given.

Author(s)

Jingyi Guo and Andrea Riebler

References

Chappell F, Raab G, Wardlaw J (2009). "When are Summary ROC Curves Appropriate for Diagnostic Meta-analyses?" Statistics in Medicine, 28(21), 2653-2668.

Arends et al.(2008) "Bivariate random effects meta-analysis of ROC curves." Medical Decision Making, 28(5), 621-638.

See Also

`SROC`, `meta4diag`

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
  require("INLA", quietly = TRUE)
  data(Catheter)
  res <- meta4diag(data = Catheter, nsample=1000)
  AUC(res, est.type="mean")
}

## End(Not run)
```

`Catheter` *The Catheter Segment Culture data.*

Description

The Cathether dataset is a collection of studies which studied semi-quantitative (19 studies) and quantitative (14 studies) catheter segment culture for the diagnosis of intravascular device-related blood stream infection.

Usage

```
data(Catheter)
```

Format

A data frame with 33 rows and 8 columns. Each row provides data for one study. The columns are "studynames", which indicates the name of studies; "type", which indicates the study quality; "prevalence", which denotes the study disease prevalence and is considered as a continuous covariate here; "TP", "FP", "TN" and "FN", which are the observations of this dataset and must be given.

References

Chu, Haitao, Hongfei Guo, and Yijie Zhou. "Bivariate Random Effects Meta-Analysis of Diagnostic Studies Using Generalized Linear Mixed Models." Medical Decision Making 30.4 (2010): 499-508.

Examples

```
  data(Catheter)
```

`crosshair` *Crosshair Plot.*

Description

Takes a `meta4diag` object produced by the function `meta4diag()` and produces a crosshair plot or adds such a plot to an existing plot. Crosshair plots go back to Phillips et al. (2010).

Usage

```
  ## S3 method for class 'meta4diag'
crosshair(x, est.type = "mean", add = FALSE, main="Crosshair Plot", xlim, ylim, col, ...)
```

Arguments

`x`	A `meta4diag` object.
`est.type`	The estimates type used to make crosshair plot. Options are "mean" and "median".
`add`	If add is TRUE, the plots are added to an existing plot, otherwise a new plot is created.
`main`	A overall title for the plot.
`xlim`	A numeric value, giving the x coordinates ranges.
`ylim`	A numeric value, giving the y coordinates ranges.
`col`	Color of cross.
`...`	graphics parameters can also be passed to this function.

Value

Besides plotting, the function returns an invisible NULL.

Author(s)

Jingyi Guo

References

Phillips, B., Stewart, L.A., & Sutton, A.J. (2010). "'Cross hairs' plots for diagnostic meta-analysis." Research Synthesis Methods, 1, 308-315.

See Also

`meta4diag`

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
  require("INLA", quietly = TRUE)
  data(Catheter)
  res <- meta4diag(data = Catheter)
  crosshair(res, est.type="mean")
}

## End(Not run)
```

`fitted` *Get the estimated values of accuracies for each study.*

Description

Takes a `meta4diag` object produced by `meta4diag()` and returns the estimated study specified accuracies, such as sensitivity, specificity, odds ratios and likelihood ratios for each study.

Usage

```
## S3 method for class 'meta4diag'
fitted(object, accuracy.type = "sens",...)
```

Arguments

`object`	A `meta4diag` object.
`accuracy.type`	A string specifying the accuracy type. Options are "sens", "TPR", "spec", "TNR", "FPR", "FNR", "LRpos", "LRneg", "RD", "LLRpos", "LLRneg", "LDOR", and "DOR". • `"sens"` and `"TPR"`: The true positive rate, known as sensitivity, $sens = \frac{TP}{TP+FN}$. • `"spec"` and `"TNR"`: The true negative rate, known as specificity, $spec = \frac{TN}{TN+FP}$. • `"FPR"`: The false positive rate, $FPR = \frac{FP}{FP+TN}$. • `"FNR"`: The false negative rate, $FNR = \frac{FN}{FN+TP}$. • `"LRpos"`: The positive likelihood ratio, $LRpos = \frac{sens}{1-spec}$. • `"LRneg"`: The negative likelihood ratio, $LRneg = \frac{1-sens}{spec}$. • `"RD"`: The risk difference, $RD = sens - (1 - spec)$. • `"DOR"`: The diagnostic odds ratio, $DOR = \frac{LRpos}{LRneg}$. • `"LLRpos"`: The log positive likelihood ratio, $LLRpos = log(\frac{sens}{1-spec})$. • `"LLRneg"`: The log negative likelihood ratio, $LLRneg = log(\frac{1-sens}{spec})$. • `"LDOR"`: The log diagnostic odds ratio, $LDOR = log(\frac{LRpos}{LRneg})$.
`...`	Arguments to be passes to methods.

Value

A data frame with the estimated mean, standard deviation and the corresponding quantiles, which are specified by user when the user runs the main function `meta4diag()`, of the accuracies of each sudy.

Author(s)

Jingyi Guo

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
  require("INLA", quietly = TRUE)
  data(Catheter)
  res <- meta4diag(data = Catheter, nsample=2000)
  fitted(res, accuracy.type="sens")
  fitted(res, accuracy.type="DOR")
}

## End(Not run)
```

`forest` *Forest plot.*

Description

Takes a `meta4diag` object produced by the function `meta4diag()` and draws a forest plot.

Usage

```
## S3 method for class 'meta4diag'
forest(x, accuracy.type="sens", est.type="mean", p.cex="scaled", p.pch=15, p.col="black",
    nameShow="right", dataShow="center", estShow="left", text.cex=1,
    shade.col="gray", arrow.col="black", arrow.lty=1, arrow.lwd=1,
    cut=TRUE, intervals=c(0.025,0.975),
    main="Forest plot", main.cex=1.5, axis.cex=1,...)
## S3 method for class 'meta4diag'
forest.grid(x, accuracy.type = "sens", est.type = "mean",
nameShow = T, dataShow = F, estShow = T, graphwidth = 1, main, xlab = "",...)
```

Arguments

`x`	A `meta4diag` object obtained by running the main function `meta4diag()`.
`accuracy.type`	A string specifying the accuracy type. Options are "sens", "TPR", "spec", "TNR", "FPR", "FNR", "LRpos", "LRneg", "RD", "LLRpos", "LLRneg", "LDOR", and "DOR".

`est.type`	The type of estimation of study specified summary points. Options are "mean" and "median".
`p.cex`	Points size of study specific estimate.
`p.pch`	Points symbol of study specific estimate.
`p.col`	Points color of study specific estimate.
`nameShow`	Boolean indicating whether the study names are shown or not. Can also be a string indicating the alignment of the study names. Options are "left", "center" and "right".
`dataShow`	Boolean indicating whether the original data is shown or not. Can also be a string indicating the position to show the original data. Options are "left", "center" and "right".
`estShow`	Boolean indicating whether the credible intervals are shown or not. Can also be a string indicating the position to show the values of credible intervals. Options are "left", "center" and "right".
`text.cex`	The size to be used for the table text.
`shade.col`	Color of shaded area.
`arrow.col`	Arrow color.
`arrow.lty`	Arrow line style.
`arrow.lwd`	Arrow line width.
`cut`	Boolean indicating the arrows should be cut or not. Or a length 2 numerical vector indicating the cut position.
`main`	An overall title for the plot.
`main.cex`	The size to be used for main titles.
`axis.cex`	The size to be used for axis annotation.
`intervals`	A numerical vector with length 2 specifying the credible intervals that is of interst. The values should be taken from the argument quantiles (see meta4diag). The first value should be smaller than 0.5 and the second value should be larger than 0.5.
`graphwidth`	A numerical value indicating the main plot device size.
`xlab`	A title for the x axis.
`...`	Arguments to be passed to methods.

Details

A forest plot is a graphical representation of a meta-analysis. It is usually accompanied by a table listing references (author and date) of the studies included in the meta-analysis. In the main device each line represents one study in the meta-analysis, plotted according to the estimated result. The estimated accuracy for each study is plotted as point and credible intervals as arrows. The size of the estimated accuracy for each study can be chosen to be propotional to the number of individuals in each study. A diamond indicates the overall summary point. Estimated accuracies and the correspoing credible intervals for each study are plotted. Together shown in the plot with the study names, the original dataset and the values of accuracies and the credible intervals.

The function `forest()` gives the user flexibility to control the plots.

Value

Besides plotting, the function returns an invisible NULL.

Author(s)

Jingyi Guo and Andrea Riebler

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
  require("INLA", quietly = TRUE)
  data(Catheter)
  res <- meta4diag(data = Catheter)
  forest(res, accuracy.type = "sens")
}

## End(Not run)
```

`funnel` *Funnel plot.*

Description

Takes a `meta4diag` object produced by the function `meta4diag()` and draws a funnel plot.

Usage

```
  ## S3 method for class 'meta4diag'
funnel(x, est.type="median",intervals=c(0.025,0.975),
  lwd=1, arrow.lty=1,col="black",
  main="Funnel Plot", xlim, ylim, ...)
```

Arguments

`x`	A `meta4diag` object obtained by running the main function `meta4diag()`.
`est.type`	The type of estimation of study specified summary points. Options are "mean" and "median".
`intervals`	A numerical vector with length 2 specifying the credible intervals that is of interst. The values should be taken from the argument quantiles (see meta4diag). The first value should be smaller than 0.5 and the second value should be larger than 0.5.
`arrow.lty`	Arrow line style.
`lwd`	Line width.
`col`	Color of cross.
`main`	An overall title for the plot.

`xlim`	A numeric value, giving the x coordinates ranges.
`ylim`	A numeric value, giving the y coordinates ranges.
`...`	Arguments to be passed to methods.

Details

A funnel plot is a graphical representation of a meta-analysis. Funnel plots are used primarily as a visual aid for detecting bias or heterogeneity, and often it is not possible to distinguish between the two. A complete collection of unbiased studies, all estimating the same underlying effect with different levels of precision, would produce a symmetric funnel shape, with increasing scatter with decreasing precision. An asymmetric funnel indicates a relationship between effect size and precision in the studies at hand. This is most likely to be due to reporting bias, to a systematic difference between smaller and larger studies, or to the presence of subsets of studies with different mean effect sizes.

The function `funnel()` gives the user flexibility to control the plots.

Value

Besides plotting, the function returns an invisible NULL.

Author(s)

Jingyi Guo and Andrea Riebler

References

Anzures-Cabrera J and Higgins J (2010). Graphical displays for meta-analysis: An overview with suggestions for practice. Research Synthesis Methods, 1, 66-80.

Examples

```
  ## Not run:
    if(requireNamespace("INLA", quietly = TRUE)){
      require("INLA", quietly = TRUE)
      data(Catheter)
      res <- meta4diag(data = Catheter)
      funnel(res, est.type = "median")
    }

## End(Not run)
```

`makeData` *Make internally used data structure.*

Description

Transform the format of the input dataset into internal use format. This function is used within the main function `meta4diag()` and can also be used as a separate function.

Usage

```
makeData(data = NULL, model.type = 1, modality = NULL, covariates = NULL)
```

Arguments

`data`	A data frame contains at least 4 columns specifying the number of True Positive(`TP`), False Negative(`FN`), True Negative(`TN`) and False Positive(`FP`). The additional columns other than `studynames` will be considered as potential covariates and the name or the column number of the potential covariates can be set in the arguments `modality` and `covariates` to use them in the model.
`model.type`	A numerical value specifying the model type, options are 1(default), 2, 3 and 4. `model.type=1` indicates that the Sensitivity(se) and Specificity(sp) will be modelled in the bivariate model, i.e. $g(se)$ and $g(sp)$ are bivariate normal distributed. `model.type=2,3,4` indicate that the Sensitivity(se) and False Negative Rate(1-sp), False Positive Rate(1-se) and Specificity(sp), False Positive Rate(1-se) and False Negative Rate(1-sp) are modelled in the bivariate model, respectively.
`modality`	A string specifying the modality variable, which is a categorical variable, such as test threshold. Default value is NULL. See also `examples`.
`covariates`	A vector specifying the continuous covariates variables, such as disease prevalence or average individual patients status of each study. Default value is NULL. See also `examples`.

Value

A constructed list with components:

`internaldata`	a data frame used as internal data in **INLA**.
`originaldata`	a data frame which is equal to the provided input data.
`covariates.setting`	a vector specified the setting of covariates, if `covariates` is given.
`modality.setting`	a vector specified the setting of modality, if `modality` is given.
`model.type`	a value specified the model type.

Author(s)

Jingyi Guo and Andrea Riebler

Examples

```
## Not run:
data(Catheter)

a = makeData(data=Catheter,model.type=1,modality="type")
b = makeData(data=Catheter,model.type=1,modality=2)
c = makeData(data=Catheter,model.type=1,covariates="prevalence")
d = makeData(data=Catheter,model.type=1,covariates=3)
e = makeData(data=Catheter,model.type=1,modality="type",covariates="prevalence")

## End(Not run)
```

`makeObject` *A function used to make a* `meta4diag` *object.*

Description

Takes an internal data list, an internal prior setting list and an `INLA` object produced by `makeData()`, `makePriors()` and `runModel()`, respectively and makes a `meta4diag` object which contains various informations for later use. This function is used in the main function `meta4diag()` and can also be used as a separate function.

Usage

```
makeObject(model, nsample=FALSE, seed=0L)
```

Arguments

`model`	An `INLA` object. Get from function `runModel()`.
`nsample`	A numerical value specifying the number of posterior samples, default is FALSE. The posterior samples are used to compute the marginals and estimates values of non-linear functions, such as log ratios and diagnostic odds ratios. If `nsample` is given, `summary.summarized.statistics`, `summary.fitted.LRpos`, `summary.fitted.LRneg`, `summary.fitted.DOR` and samples of $E(se)$, $E(sp)$, $E(1-se)$ and $E(1-sp)$ will be returned.
`seed`	A numerical value specifying the random seed to control the RNG for generating posterior samples if nsample > 0. If you want reproducible results, you ALSO need to control the seed for the RNG in R by controlling the variable .Random.seed or using the function set.seed.

Value

`makeObject` returns a `meta4diag` object with components:

`data`	The provided input data.
`outdata`	The internal data that could be used in **INLA** from function `makeData()`.

`priors.density`	Prior distributions for the variance components and correlation from function `makePriors()`.
`names.fitted`	Names of the jointly modelled accuracies in the model. For example, se and sp or (1-se) and sp.
`cpu.used`	The cpu time used for running the model.
`call`	The matched call.
`summary.fixed`	Matrix containing the mean and standard deviation (plus, possibly quantiles) of the fixed effects of the model.
`marginals.fixed`	A list containing the posterior marginal densities of the fixed effects of the model.
`summary.expected.(...).accuracy`	Matrix containing the mean and standard deviation (plus, possibly quantiles) of the mean of accuracies transformed with the link function, i.e. E(g(Se)), E(g(Sp)), E(g(1-Se)) and E(g(1-Sp)). Dynamic name for this output. (...) indicates the name of link function used in `runModel()`, i.e. if link function is "logit", the full name of this output is "summary.expected.logit.accuracy".
`marginals.expected.(...).accuracy`	A list containing the posterior marginal densities of the mean of accuracies transformed with the link function, i.e. E(g(Se)), E(g(Sp)), E(g(1-Se)) and E(g(1-Sp)). Dynamic name for this output. (...) indicates the name of link function used in `runModel()`, i.e. if link function is "logit", the full name of this output is "marginals.expected.logit.accuracy".
`summary.expected.accuracy`	Matrix containing the mean and standard deviation (plus, possibly quantiles) of the mean of the accuracies, i.e. E(Se), E(Sp), E(1-Se) and E(1-Sp).
`marginals.expected.accuracy`	A list containing the posterior marginal densities of of the mean of the accuracies, i.e. E(Se), E(Sp), E(1-Se) and E(1-Sp).
`summary.hyperpar`	A matrix containing the mean and sd (plus, possibly quantiles) of the hyperparameters of the model.
`marginals.hyperpar`	A list containing the posterior marginal densities of the hyperparameters of the model.
`correlation.expected.(...).accuracy`	A correlation matrix between the mean of the accuracies transformed with the link function. Dynamic name for this output. (...) indicates the name of link function used in `runModel()`.
`covariance.expected.(...).accuracy`	A covariance matrix between the mean of the accuracies transformed with the link function. Dynamic name for this output. (...) indicates the name of link function used in `runModel()`.
`summary.predictor.(...)`	A matrix containing the mean and sd (plus, possibly quantiles) of the linear predictors one transformed accuracy in the model. The accuracy type depends on

	the model type. See argument `model.type`. For example, the possible accuracy type could be $g(se)$, $g(sp)$, (se) or (sp), where $g()$ is the link function.
`marginals.predictor.(...)`	A list containing the posterior marginals of the linear predictors of one transformed accuracy in the model. The accuracy type depends on the model type. See argument `model.type`. For example, the possible accuracy type could be $g(se)$, $g(sp)$, (se) or (sp), where $g()$ is the link function.
`misc`	Some other settings that maybe useful retruned by **meta4diag**.
`dic`	The deviance information criteria and effective number of parameters.
`cpo`	A list of three elements: `cpo$cpo` are the values of the conditional predictive ordinate (CPO), `cpo$pit` are the values of the probability integral transform (PIT) and `cpo$failure` indicates whether some assumptions are violated. In short, if cpo$failure[i] > 0 then some assumption is violated, the higher the value (maximum 1) the more seriously.
`waic`	A list of two elements: `waic$waic` is the Watanabe-Akaike information criteria, and `waic$p.eff` is the estimated effective number of parameters.
`mlik`	The log marginal likelihood of the model
`inla.result`	A `INLA` object that from function `runModel()` which implements INLA.
`samples.fixed`	A matrix of the fixed effects samples if `nsample` is given.
`samples.hyperpar`	A matrix of the hyperparameter samples if `nsample` is given.
`samples.overall.Se`	A matrix containing the mean and sd (plus, possibly quantiles) of overall sensitivity samples if `nsample` is given.
`samples.overall.Sp`	A matrix containing the mean and sd (plus, possibly quantiles) of overall specificity samples if `nsample` is given.
`summary.overall.statistics`	A matrix containing the mean and sd (plus, possibly quantiles) of mean positive and negative likelihood ratios and mean diagnostic odds ratios if `nsample` is given.
`samples.study.specific.Se`	A matrix containing the mean and sd (plus, possibly quantiles) of study specific sensitivity samples if `nsample` is given.
`samples.study.specific.Sp`	A matrix containing the mean and sd (plus, possibly quantiles) of study specific specificity samples if `nsample` is given.
`summary.study.specific.LRpos`	A matrix containing the mean and sd (plus, possibly quantiles) of positive likelihood ratios for each study if `nsample` is given.
`summary.study.specific.LRneg`	A matrix containing the mean and sd (plus, possibly quantiles) of negative likelihood ratios for each study if `nsample` is given.

`summary.study.specific.DOR`	A matrix containing the mean and sd (plus, possibly quantiles) of diagnostic odds ratios for each study if `nsample` is given.
`summary.study.specific.RD`	A matrix containing the mean and sd (plus, possibly quantiles) of risk difference for each study if `nsample` is given.
`summary.study.specific.LDOR`	A matrix containing the mean and sd (plus, possibly quantiles) of log diagnostic odds ratios for each study if `nsample` is given.
`summary.study.specific.LLRpos`	A matrix containing the mean and sd (plus, possibly quantiles) of log positive likelihood ratios for each study if `nsample` is given.
`summary.study.specific.LLRneg`	A matrix containing the mean and sd (plus, possibly quantiles) of log negative likelihood ratios for each study if `nsample` is given.

Author(s)

Jingyi Guo

See Also

`makeData`, `makePriors`, `runModel`, `meta4diag`

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
  require("INLA", quietly = TRUE)
  data(Catheter)
  outdata = makeData(Catheter)
  outpriors = makePriors()
  model = runModel(outdata=outdata, outpriors=outpriors, link="logit")
  res = makeObject(outdata, outpriors, model, nsample=2000)
}

## End(Not run)
```

`makePriors` *Priors for internal use.*

Description

Prepare the given prior settings for internal use in **INLA**. The input of the function `makePriors` are prior settings for variances and correlation or the full covariance matrix, these are then transformed to the internal parameterisation that **INLA** requires. This function is used internally in the main function `meta4diag()` but can also be used as a separate function.

Usage

```
makePriors(var.prior = "PC", var2.prior="PC", cor.prior = "PC",
  var.par = c(3, 0.05), var2.par, cor.par = c(1,-0.1,0.5,-0.95,0.05,0.95,0.05),
  wishart.par = c(4,1,1,0), init = c(0.01, 0.01, -0.1))
```

Arguments

`var.prior` A string specifying the prior density for the first variance component, options are "PC" for penalised complexity prior, "Invgamma" for inverse gamma prior, "Tnormal" for truncated normal prior, "Unif" for uniform prior which allow the standard deviation uniformaly distributed on [0,1000], "Hcauchy" for half-cauchy prior and "table" for user specific prior. `var.prior` can also be set to "Invwishart" for inverse wishart prior for covariance matrix. When `var.prior="Invwishart"`, no matter what `var2.prior` and `cor.prior` are given, the inverse Wishart prior covariance matrix is used for covariance matrix and the `wishart.par` must be given. Of note, the values of this argument is not case sensitive. The definition of the priors is as following,

- `var.prior="Invgamma"`: This is a prior for a variance σ^2. The inverse gamma prior has density,

  $$\pi(\sigma^2) = \frac{1}{\Gamma(a)b^a}(\sigma^2)^{-a-1}exp(-\frac{1}{b\sigma^2}),$$

  for $\sigma^2 > 0$ where: $a > 0$ is the shape parameter, $b > 0$ is the rate (1/scale) parameter. The parameters are here `c(a, b)`, see arguments `var.par`.
- `var.prior="Tnormal"`: This is a prior for a variance σ^2 and defined as follows. The standard deviation $\sigma = \sqrt{\sigma^2}$ is Gaussian distributed with mean m and variance v but truncated to be positive. The parameters are here `c(m, v)`, see arguments `var.par`.
- `var.prior="PC"`: This is a prior for a variance σ^2 and defined as follows. The left tail of the distribution of standard deviation σ has behavior

  $$P(\sigma > u) = \alpha,$$

  which means it is unlikely that the standard deviation σ to be larger than a value u with probability α. The parameters are here `c(u, alpha)`, see arguments `var.par`.
- `var.prior="Hcauchy"`: This is a prior for a variance σ^2 and defined as follows. The standard deviation $\sigma = \sqrt{\sigma^2}$ is half-Cauchy distributed with density,

  $$\pi(\sigma) = \frac{2\gamma}{\pi(\sigma^2 + \gamma^2)},$$

  where $\gamma > 0$ is the rate parameter. The parameters are here `c(gamma)`, see arguments `var.par`.
- `var.prior="Unif"`: This is a prior for a variance σ^2 and defined as follows. The standard deviation $\sigma = \sqrt{\sigma^2}$ is uniform distributed on $(0, \infty)$. No parameters need to be given for this prior, see arguments `var.par`.

- `var.prior="Table"`: This is a prior for a variance σ^2 and defined as follows. Users have to specify a data.frame with 2 columns, one indicates the values of σ^2 and the other one indicates the values of $\pi(\sigma^2)$. The parameters are this data frame, see arguments `var.par`.
- `var.prior="Invwishart"`: Instead of specifying separate prior distributions for the variance and correlation parameters we could also assume that the covariance matrix Σ

  $$\Sigma \sim Wishart_p^{-1}(\nu, R), p = 2,$$

  where the Wishart distribution has density

  $$\pi(\Sigma) = \frac{|R|^{\frac{\nu}{2}}}{2^{\frac{p\nu}{2}}\Gamma_p(\frac{\nu}{2})}|\Sigma|^{-\frac{\nu+p+1}{2}} exp(-\frac{1}{2}Trace(\frac{R}{\Sigma})), \nu > p + 1,$$

  Then,

  $$E(\Sigma) = \frac{R}{\nu - p - 1}.$$

  The parameters are $\nu, R_{11}, R_{22} and R_{12}$, where

  $$R = \left(\begin{array}{cc} R_{11} & R_{12} \\ R_{21} & R_{22} \end{array} \right)$$

  The parameters are here `c(nu, R11, R22, R12)`, see arguments `var.par`.

`var2.prior` See `var.prior`.

`cor.prior` A string specifying the prior density for the correlation, options are "PC" for penalised complexity prior, "Invgamma" for inverse gamma prior, "Beta" for beta prior and "table" for user specific prior. `cor.prior` can also be set to "Invwishart" for inverse wishart prior for covariance matrix. When `cor.prior="Invwishart"`, no matter what `var.prior` and `var2.prior` are given, the inverse Wishart prior for covariance matrix is used and the `wishart.par` must be given. Of note, the values of this argument is not case sensitive. The definition of the priors is as following,

- `cor.prior="Normal"`: This is a prior for a correlation ρ and defined as follows. The correlation parameter is constrained to $[-1, 1]$. We reparameterise the correlation parameter ρ using Fisher's z-transformation as

  $$\rho^\star = logit(\frac{\rho + 1}{2}),$$

  which assumes values over the whole real line and assign the following prior distribution to ρ,

  $$\rho \sim Gaussian(\mu, \sigma^2).$$

  The prior variance of 2.5 and prior mean of 0 corresponds, roughly, to a uniform prior on $[-1, 1]$ for ρ . The parameters are here `c(mean, variance)`, see arguments `cor.par`.

- `cor.prior="PC"`: This is a prior for a correlation ρ and defined as follows. The prior is defined around at a reference point with value ρ_0. To define the density behavior, three `strategy` can be applied. The first strategy is to define the left tail behavior and the density weight on the left-hand side of the reference point ρ_0,
  $$P(\rho < u_1) = a_1,$$
  and
  $$P(\rho < \rho_0) = \omega,$$
  which means it is unlikely that the value of ρ is smaller than a small value u_1 with probability a_1 and the probability that ρ is smaller than ρ_0 is ω. The parameters for the first strategy are here `c(1, rho0, omega, u1, a1, NA, NA)`, see arguments `cor.par`. The second strategy is to define the right tail behavior and the density weight on the left-hand side of the reference point ρ_0,
  $$P(\rho > u_2) = a_2,$$
  and
  $$P(\rho < \rho_0) = \omega,$$
  which means it is unlikely that the value of ρ is larger than a big value u_2 with probability a_2 and the probability that ρ is smaller than ρ_0 is ω. The parameters for the second strategy are here `c(2, rho0, omega, NA, NA, u2, a2)`, see arguments `cor.par`. The third strategy is to define both tail behaviors,
  $$P(\rho < u_1) = a_1,$$
  and
  $$P(\rho > u_2) = a_2.$$
  The parameters for the third strategy are here `c(3, rho0, NA, u1, a1, u2, a2)`, see arguments `cor.par`. The parameters of the PC prior for the correlation here is `c(strategy, rho0, omega, u1, a1, u2, a2)`, see arguments `cor.par`.
- `cor.prior="Beta"`: This is a prior for a correlation ρ and defined as follows. The correlation parameter ρ has a $Beta(a, b)$ distribution scaled to have domain in $(-1, 1)$:
  $$\pi(\rho) = 0.5\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\rho^{a-1}(1-\rho)^{b-1}$$
  , where $a, b > 0$ are the shape parameter. The parameters are here `c(a, b)`, see arguments `cor.par`.
- `cor.prior="Table"`: This is a prior for a correlation ρ and defined as follows. Users have to specify the data.frame with 2 columns, one indicates the values of ρ and the other one indicates the values of $\pi(\rho)$. The parameters are this data frame, see arguments `cor.par`.
- `cor.prior="Invwishart"`: Instead of specifying separate prior distributions for the hyperparameters we could also assume that the covariance matrix Σ
  $$\Sigma \sim Wishart_p^{-1}(\nu, R), p = 2,$$

where the inverse Wishart distribution has density

$$\pi(\Sigma) = \frac{|R|^{\frac{\nu}{2}}}{2^{\frac{p\nu}{2}}\Gamma_p(\frac{\nu}{2})}|\Sigma|^{-\frac{\nu+p+1}{2}}exp(-\frac{1}{2}Trace(\frac{R}{\Sigma})), \nu > p+1,$$

Then,

$$E(\Sigma) = \frac{R}{\nu - p - 1}.$$

The parameters are ν, R_{11}, $R_{22} and R_{12}$, where

$$R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix}$$

The parameters are here `c(nu, R11, R22, R12)`, see arguments `cor.par`.

`var.par`	A numerical vector specifying the parameter of the prior density for the first variance component. - `var.par=c(rate, shape)` when `var.prior="Invgamma"`. - `var.par=c(u, alpha)` when `var.prior="PC"`. - `var.par=c(m, v)` when `var.prior="Tnormal"`. - `var.par=c(gamma)` when `var.prior="Hcauchy"`. - `var.par=c()` when `var.prior="Unif"`. - `var.par` is a data frame with 2 columns, one indicates the values of σ^2 and the other one indicates the values of $\pi(\sigma^2)$ when `var.prior="Table"`. - `var.par` doesn't need to be given when `var.prior="Invwishart"`. See also argument `var.prior`.
`var2.par`	A numerical vector specifying the parameter of the prior density for the second variance component. If not given, function will copy the setting for the first variance component. The definition of the priors is the same as for `var.par`.
`cor.par`	A numerical vector specifying the parameter of the prior density for the correlation. See also `examples`. - `cor.par=c(mean, variance)` when `cor.prior="normal"`. - `cor.par=c(strategy, rho0, omega, u1, a1, u2, a2)` when `cor.prior="PC"`. - `cor.par=c(a, b)` when `var.prior="beta"`. - `cor.par` is a data frame with 2 columns, one indicates the values of ρ and the other one indicates the values of $\pi(\rho)$ when `cor.prior="Table"`. - `cor.par` doesn't need to be given when `cor.prior="Invwishart"`. See also argument `cor.prior`.
`wishart.par`	A numerical vector specifying the parameter of the prior density for the covariance matrix. `wishart.par` must be given and `wishart.par=c(nu, R11, R22, R12)` when any of `var.prior`, `var2.prior` or `cor.prior` is `"Invwishart"`. See also `examples`.
`init`	A numerical vector specifying the initial value of the first variance, the second variance and correlation.

Value

A list of prior settings with the components:

`prec1`	a list of prior settings for the first log precision (the log inverse of the first variance in the model).
`prec2`	a list of prior settings for the second log precision (the log inverse of the second variance in the model).
`cor`	a list of prior settings for the transformed correlation (some functions of correlation in the model).
`lambdas`	a vector of rate parameters for the PC correlation if `cor.prior="PC"`.
`density`	a list of prior densities for precisions and correlations.
`original.setting`	a list of input prior settings.
`wishart.flag`	Boolean indicating whether a inverse Wishart prior is setting or not.

Author(s)

Jingyi Guo and Andrea Riebler

References

Simpson DP, Martins TG, Riebler A, Fuglstad GA, Rue H, Sorbye SH (2014) Penalised Model Component Complexity: A principled, Practical Approach to Constructing Priors. Arxiv e-prints. 1403.4630

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
require("INLA", quietly = TRUE)
makePriors(var.prior = "PC", cor.prior = "PC", var.par = c(3, 0.05),
  cor.par = c(1, -0.1, 0.5, -0.95, 0.05, NA, NA))
makePriors(var.prior = "PC", cor.prior = "PC", var.par = c(3, 0.05),
  cor.par = c(2, -0.1, 0.5, NA, NA, 0.95, 0.05))
makePriors(var.prior = "PC", cor.prior = "PC", var.par = c(3, 0.05),
  cor.par = c(3, -0.1, NA, -0.95, 0.05, 0.95, 0.05))
makePriors(var.prior = "invgamma", cor.prior = "normal",
  var.par = c(0.25, 0.025), cor.par = c(0, 5))
makePriors(var.prior = "invwishart", wishart.par=c(4,1,2,0.1))
}

## End(Not run)
```

`meta4diag` *Function to analyse diagnostic meta-analysis with Bayesian methods using INLA.*

Description

Estimate a Bayesian bivariate hierarchical model fitted within INLA.

Usage

```
meta4diag(data=NULL, model.type = 1,
         var.prior = "Invgamma", var2.prior = "Invgamma", cor.prior = "Normal",
          var.par = c(0.25, 0.025), var2.par, cor.par = c(0,5),
          wishart.par = c(4, 1, 1, 0),
          init = c(0.01,0.01,0), link="logit", quantiles=c(0.025,0.5,0.975),
          modality = NULL, covariates = NULL,
          verbose = FALSE, nsample=FALSE,num.threads = 1, seed=0L)
```

Arguments

`data`	A data frame contains at least 4 columns specifying the number of True Positive(`TP`), False Negative(`FN`), True Negative(`TN`) and False Positive(`FP`). The additional columns other than `studynames` will be considered as potential covariates and the name or the column number of the potential covariates can be set in the arguments `modality` and `covariates` to use them in the model.
`model.type`	A numerical value specifying the model type, options are 1(default), 2, 3 and 4. `model.type=1` indicates that the Sensitivity(se) and Specificity(sp) will be modelled in the bivariate model, i.e. $g(se)$ and $g(sp)$ are bivariate normal distributed. `model.type=2,3,4` indicate that the Sensitivity(se) and False Negative Rate(1-sp), False Positive Rate(1-se) and Specificity(sp), False Positive Rate(1-se) and False Negative Rate(1-sp) are modelled in the bivariate model, respectively.
`var.prior`	A string specifying the prior density for the first variance component, options are "PC" for penalised complexity prior, "Invgamma" for inverse gamma prior, "Tnormal" for truncated normal prior, "Unif" for uniform prior which allow the standard deviation uniformaly distributed on [0,1000], "Hcauchy" for half-cauchy prior and "table" for user specific prior. `var.prior` can also be set to "Invwishart" for inverse wishart prior for covariance matrix. When `var.prior="Invwishart"`, no matter what `var2.prior` and `cor.prior` are given, the inverse Wishart prior covariance matrix is used for covariance matrix and the `wishart.par` must be given. The definition of the priors is as following, • `var.prior="Invgamma"`: This is a prior for a variance σ^2. The inverse gamma prior has density, $\pi(\sigma^2) = \frac{1}{\Gamma(a)b^a}(\sigma^2)^{-a-1}exp(-\frac{1}{b\sigma^2}),$ for $\sigma^2 > 0$ where: $a > 0$ is the shape parameter, $b > 0$ is the rate (1/scale) parameter. The parameters are here `c(a, b)`, see arguments `var.par`.

- `var.prior="Tnormal"`: This is a prior for a variance σ^2 and defined as follows. The standard deviation $\sigma = \sqrt{\sigma^2}$ is Gaussian distributed with mean m and variance v but truncated to be positive. The parameters are here `c(m, v)`, see arguments `var.par`.
- `var.prior="PC"`: This is a prior for a variance σ^2 and defined as follows. The left tail of the distribution of standard deviation σ has behavior

  $$P(\sigma > u) = \alpha,$$

  which means it is unlikely that the standard deviation σ to be larger than a value u with probability α. The parameters are here `c(u, alpha)`, see arguments `var.par`.
- `var.prior="HCauchy"`: This is a prior for a variance σ^2 and defined as follows. The standard deviation $\sigma = \sqrt{\sigma^2}$ is half-Cauchy distributed with density,

  $$\pi(\sigma) = \frac{2\gamma}{\pi(\sigma^2 + \gamma^2)},$$

  where $\gamma > 0$ is the rate parameter. The parameters are here `c(gamma)`, see arguments `var.par`.
- `var.prior="Unif"`: This is a prior for a variance σ^2 and defined as follows. The standard deviation $\sigma = \sqrt{\sigma^2}$ is uniform distributed on $(0, \infty)$. No parameters need to be given for this prior, see arguments `var.par`.
- `var.prior="Table"`: This is a prior for a variance σ^2 and defined as follows. Users have to specify a data.frame with 2 columns, one indicates the values of σ^2 and the other one indicates the values of $\pi(\sigma^2)$. The parameters are this data frame, see arguments `var.par`.
- `var.prior="Invwishart"`: Instead of specifying separate prior distributions for the variance and correlation parameters we could also assume that the covariance matrix Σ

  $$\Sigma \sim Wishart_p^{-1}(\nu, R), p = 2,$$

  where the Wishart distribution has density

  $$\pi(\Sigma) = \frac{|R|^{\frac{\nu}{2}}}{2^{\frac{p\nu}{2}}\Gamma_p(\frac{\nu}{2})}|\Sigma|^{-\frac{\nu+p+1}{2}} exp(-\frac{1}{2}Trace(\frac{R}{\Sigma})), \nu > p + 1,$$

  Then,

  $$E(\Sigma) = \frac{R}{\nu - p - 1}.$$

  The parameters are $\nu, R_{11}, R_{22} and R_{12}$, where

  $$R = \left(\begin{array}{cc} R_{11} & R_{12} \\ R_{21} & R_{22} \end{array} \right)$$

  The parameters are here `c(nu, R11, R22, R12)`, see arguments `var.par`.

`var2.prior` See `var.prior`.

`cor.prior` A string specifying the prior density for the correlation, options are "PC" for penalised complexity prior, "Invgamma" for inverse gamma prior, "beta" for beta prior and "table" for user specific prior. `cor.prior` can also be set to "Invwishart" for inverse wishart prior for covariance matrix. When `cor.prior="Invwishart"`, no matter what `var.prior` and `var2.prior` are given, the inverse Wishart prior for covariance matrix is used and the `wishart.par` must be given. The definition of the priors is as following,

- `cor.prior="Normal"`: This is a prior for a correlation ρ and defined as follows. The correlation parameter is constrained to $[-1, 1]$. We reparameterise the correlation parameter ρ using Fisher's z-transformation as

  $$\rho^{\star} = logit(\frac{\rho + 1}{2}),$$

  which assumes values over the whole real line and assign the following prior distribution to ρ,

  $$\rho \sim Gaussian(\mu, \sigma^2).$$

  The prior variance of 2.5 and prior mean of 0 corresponds, roughly, to a uniform prior on $[-1, 1]$ for ρ . The parameters are here `c(mean, variance)`, see arguments `cor.par`.
- `cor.prior="PC"`: This is a prior for a correlation ρ and defined as follows. The prior is defined around at a reference point with value ρ_0. To define the density behavior, three `strategy` can be applied. The first strategy is to define the left tail behavior and the density weight on the left-hand side of the reference point ρ_0,

  $$P(\rho < u_1) = a_1,$$

  and

  $$P(\rho < \rho_0) = \omega,$$

  which means it is unlikely that the value of ρ is smaller than a small value u_1 with probability a_1 and the probability that ρ is smaller than ρ_0 is ω. The parameters for the first strategy are here `c(1, rho0, omega, u1, a1, NA, NA)`, see arguments `cor.par`. The second strategy is to define the right tail behavior and the density weight on the left-hand side of the reference point ρ_0,

  $$P(\rho > u_2) = a_2,$$

  and

  $$P(\rho < \rho_0) = \omega,$$

  which means it is unlikely that the value of ρ is larger than a big value u_2 with probability a_2 and the probability that ρ is smaller than ρ_0 is ω. The parameters for the second strategy are here `c(2, rho0, omega, NA, NA, u2, a2)`, see arguments `cor.par`. The third strategy is to define both tail behaviors,

  $$P(\rho < u_1) = a_1,$$

  and

  $$P(\rho > u_2) = a_2.$$

The parameters for the third strategy are here `c(3, rho0, NA, u1, a1, u2, a2)`, see arguments `cor.par`. The parameters of the PC prior for the correlation here is `c(strategy, rho0, omega, u1, a1, u2, a2)`, see arguments `cor.par`.

- `cor.prior="beta"`: This is a prior for a correlation ρ and defined as follows. The correlation parameter ρ has a $Beta(a, b)$ distribution scaled to have domain in $(-1, 1)$:

  $$\pi(\rho) = 0.5\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\rho^{a-1}(1-\rho)^{b-1}$$

  , where $a, b > 0$ are the shape parameter. The parameters are here `c(a, b)`, see arguments `cor.par`.
- `cor.prior="Table"`: This is a prior for a correlation ρ and defined as follows. Users have to specify the data.frame with 2 columns, one indicates the values of ρ and the other one indicates the values of $\pi(\rho)$. The parameters are this data frame, see arguments `cor.par`.
- `cor.prior="Invwishart"`: Instead of specifying separate prior distributions for the hyperparameters we could also assume that the covariance matrix Σ

  $$\Sigma \sim Wishart_p^{-1}(\nu, R), p = 2,$$

  where the inverse Wishart distribution has density

  $$\pi(\Sigma) = \frac{|R|^{\frac{\nu}{2}}}{2^{\frac{p\nu}{2}}\Gamma_p(\frac{\nu}{2})}|\Sigma|^{-\frac{\nu+p+1}{2}}exp(-\frac{1}{2}Trace(\frac{R}{\Sigma})), \nu > p+1,$$

  Then,

  $$E(\Sigma) = \frac{R}{\nu - p - 1}.$$

  The parameters are $\nu, R_{11}, R_{22} and R_{12}$, where

  $$R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix}$$

  The parameters are here `c(nu, R11, R22, R12)`, see arguments `cor.par`.

`var.par` A numerical vector specifying the parameter of the prior density for the first variance component.

- `var.par=c(rate, shape)` when `var.prior="Invgamma"`.
- `var.par=c(u, alpha)` when `var.prior="PC"`.
- `var.par=c(m, v)` when `var.prior="Tnormal"`.
- `var.par=c(gamma)` when `var.prior="Hcauchy"`.
- `var.par=c()` when `var.prior="Unif"`.
- `var.par` is a data frame with 2 columns, one indicates the values of σ^2 and the other one indicates the values of $\pi(\sigma^2)$ when `var.prior="Table"`.
- `var.par` doesn't need to be given when `var.prior="Invwishart"`.

See also argument `var.prior`.

`var2.par`	A numerical vector specifying the parameter of the prior density for the second variance component. If not given, function will copy the setting for the first variance component. The definition of the priors is the same as for `var.par`.
`cor.par`	A numerical vector specifying the parameter of the prior density for the correlation. See also `examples`. • `cor.par=c(mean, variance)` when `cor.prior="normal"`. • `cor.par=c(strategy, rho0, omega, u1, a1, u2, a2)` when `cor.prior="PC"`. • `cor.par=c(a, b)` when `var.prior="beta"`. • `cor.par` is a data frame with 2 columns, one indicates the values of ρ and the other one indicates the values of $\pi(\rho)$ when `cor.prior="Table"`. • `cor.par` doesn't need to be given when `cor.prior="Invwishart"`. See also argument `cor.prior`.
`wishart.par`	A numerical vector specifying the parameter of the prior density for the covariance matrix. `wishart.par` must be given and `wishart.par=c(nu, R11, R22, R12)` when any of `var.prior`, `var2.prior` or `cor.prior` is "Invwishart". See also `examples`.
`init`	A numerical vector specifying the initial value of the first variance, the second variance and correlation.
`link`	A string specifying the link function used in the model. Options are "logit", "probit" and "cloglog".
`quantiles`	A vector of quantiles, p(0), p(1),... to compute for each posterior marginal. The function returns, for each posterior marginal, the values x(0), x(1),... such that $Prob(X < x) = p.$ The default value are c(0.025, 0.5, 0.975). Not matter what other values are going to be given, the estimates for these 3 quantiles are always returned.
`verbose`	Boolean (default:FALSE) indicating whether the program should run in a verbose model.
`modality`	A string specifying the modality variable, which is a categorical variable, such as test threshold. Default value is NULL. See also `examples`. At the moment, only one categorical covariate variable can be used in the model.
`covariates`	A vector, which could be either the name of columns or the number of columns, specifying the continuous covariates variables, such as disease prevalence or average individual patients status of each study. Default value is NULL. See also `examples`.
`nsample`	A numerical value specifying the number of posterior samples, default is 5000. The posterior samples are used to compute the marginals and estimates values of non-linear functions, such as log radios and diagnostic odds radios. If `nsample` is given, `summary.summarized.statistics`, `summary.fitted.LRpos`, `summary.fitted.LRneg`, `summary.fitted.DOR` and samples of $E(se)$, $E(sp)$, $E(1-se)$ and $E(1-sp)$ will be returned.
`num.threads`	Maximum number of threads the inla-program will use. xFor Windows this defaults to 1, otherwise its defaults to NULL (for which the system takes over control).

`seed`	A numerical value specifying the random seed to control the RNG for generating posterior samples if nsample > 0. If you want reproducible results, you ALSO need to control the seed for the RNG in R by controlling the variable .Random.seed or using the function set.seed.

Details

The bivariate model has two levels, in the first level, the observed number of individuals in a specific group in a 2 by 2 table is binomial distributed, for example, the numbers of individuals in the group of true positive and true negative of a study i are modelled jointly,

$$TP_i|Se_i \sim Binomial(TP_i + FN_i, Se_i)$$

$$TN_i|Sp_i \sim Binomial(TN_i + FP_i, Sp_i)$$

In the second level, two transformed accuracies with some link function (see argument `link`) are bivariate Gaussian distributed, continuous with the previous example,

$$g(Se_i) = \mu + V_i\alpha + \phi_i$$

$$g(Sp_i) = \nu + U_i\beta + \psi_i$$

where ϕ_i and ψ_i are bivariate Gaussian distributed with mean 0 and covariance matrix Σ. The se and sp in the example could be changed to se and $(1-sp)$, $(1-se)$ and sp or $(1-se)$ and $(1-sp)$, see argument `model.type`.

The function `meta4diag()` depends on four internal functions which are also given in the **meta4diag** package in order to flexibly implement a list of dataset with the same prior setting. The four internal functions are `makeData()`, `makePriors()`, `runModel()` and `makeObject()`. Details can be seen the corresponding documentations and examples.

After running the function `meta4diag()`, a `meta4diag` object will be returned which contains various estimated results for later analysis, such as the posterior marginals, estimated value, standard deviations and the coresponding quaniles of the accuracies. See `Values`.

Value

`meta4diag` returns a `meta4diag` object with components:

`data`	The provided input data.
`outdata`	The internal data that could be used in **INLA** from function `makeData()`.
`priors.density`	Prior distributions for the variance components and correlation from function `makePriors()`.
`names.fitted`	Names of the jointly modelled accuracies in the model. For example, se and sp or (1-se) and sp.
`cpu.used`	The cpu time used for running the model.
`call`	The matched call.
`summary.fixed`	Matrix containing the mean and standard deviation (plus, possibly quantiles) of the fixed effects of the model.

`marginals.fixed`	A list containing the posterior marginal densities of the fixed effects of the model.
`summary.expected.(...).accuracy`	Matrix containing the mean and standard deviation (plus, possibly quantiles) of the mean of accuracies transformed with the link function, i.e. E(g(Se)), E(g(Sp)), E(g(1-Se)) and E(g(1-Sp)). Dynamic name for this output. (...) indicates the name of link function used in `runModel()`, i.e. if link function is "logit", the full name of this output is "summary.expected.logit.accuracy".
`marginals.expected.(...).accuracy`	A list containing the posterior marginal densities of the mean of accuracies transformed with the link function, i.e. E(g(Se)), E(g(Sp)), E(g(1-Se)) and E(g(1-Sp)). Dynamic name for this output. (...) indicates the name of link function used in `runModel()`, i.e. if link function is "logit", the full name of this output is "marginals.expected.logit.accuracy".
`summary.expected.accuracy`	Matrix containing the mean and standard deviation (plus, possibly quantiles) of the mean of the accuracies, i.e. E(Se), E(Sp), E(1-Se) and E(1-Sp).
`marginals.expected.accuracy`	A list containing the posterior marginal densities of of the mean of the accuracies, i.e. E(Se), E(Sp), E(1-Se) and E(1-Sp).
`summary.hyperpar`	A matrix containing the mean and sd (plus, possibly quantiles) of the hyperparameters of the model.
`marginals.hyperpar`	A list containing the posterior marginal densities of the hyperparameters of the model.
`correlation.expected.(...).accuracy`	A correlation matrix between the mean of the accuracies transformed with the link function. Dynamic name for this output. (...) indicates the name of link function used in `runModel()`.
`covariance.expected.(...).accuracy`	A covariance matrix between the mean of the accuracies transformed with the link function. Dynamic name for this output. (...) indicates the name of link function used in `runModel()`.
`summary.predictor.(...)`	A matrix containing the mean and sd (plus, possibly quantiles) of the linear predictors one transformed accuracy in the model. The accuracy type depends on the model type. See argument `model.type`. For example, the possible accuracy type could be $g(se)$, $g(sp)$, (se) or (sp), where $g()$ is the link function.
`marginals.predictor.(...)`	A list containing the posterior marginals of the linear predictors of one transformed accuracy in the model. The accuracy type depends on the model type. See argument `model.type`. For example, the possible accuracy type could be $g(se)$, $g(sp)$, (se) or (sp), where $g()$ is the link function.
`misc`	Some other settings that maybe useful retruned by **meta4diag**.
`dic`	The deviance information criteria and effective number of parameters.

`cpo`	A list of three elements: `cpo$cpo` are the values of the conditional predictive ordinate (CPO), `cpo$pit` are the values of the probability integral transform (PIT) and `cpo$failure` indicates whether some assumptions are violated. In short, if cpo$failure[i] > 0 then some assumption is violated, the higher the value (maximum 1) the more seriously.
`waic`	A list of two elements: `waic$waic` is the Watanabe-Akaike information criteria, and `waic$p.eff` is the estimated effective number of parameters.
`mlik`	The log marginal likelihood of the model
`inla.result`	A `INLA` object that from function `runModel()` which implements INLA.
`samples.fixed`	A matrix of the fixed effects samples if `nsample` is given.
`samples.hyperpar`	A matrix of the hyperparameter samples if `nsample` is given.
`samples.overall.Se`	A matrix containing the mean and sd (plus, possibly quantiles) of overall sensitivity samples if `nsample` is given.
`samples.overall.Sp`	A matrix containing the mean and sd (plus, possibly quantiles) of overall specificity samples if `nsample` is given.
`summary.overall.statistics`	A matrix containing the mean and sd (plus, possibly quantiles) of mean positive and negative likelihood ratios and mean diagnostic odds ratios if `nsample` is given.
`samples.study.specific.Se`	A matrix containing the mean and sd (plus, possibly quantiles) of study specific sensitivity samples if `nsample` is given.
`samples.study.specific.Sp`	A matrix containing the mean and sd (plus, possibly quantiles) of study specific specificity samples if `nsample` is given.
`summary.study.specific.LRpos`	A matrix containing the mean and sd (plus, possibly quantiles) of positive likelihood ratios for each study if `nsample` is given.
`summary.study.specific.LRneg`	A matrix containing the mean and sd (plus, possibly quantiles) of negative likelihood ratios for each study if `nsample` is given.
`summary.study.specific.DOR`	A matrix containing the mean and sd (plus, possibly quantiles) of diagnostic odds ratios for each study if `nsample` is given.
`summary.study.specific.RD`	A matrix containing the mean and sd (plus, possibly quantiles) of risk difference for each study if `nsample` is given.
`summary.study.specific.LDOR`	A matrix containing the mean and sd (plus, possibly quantiles) of log diagnostic odds ratios for each study if `nsample` is given.
`summary.study.specific.LLRpos`	A matrix containing the mean and sd (plus, possibly quantiles) of log positive likelihood ratios for each study if `nsample` is given.

`summary.study.specific.LLRneg`
A matrix containing the mean and sd (plus, possibly quantiles) of log negative likelihood ratios for each study if `nsample` is given.

Author(s)

Jingyi Guo and Andrea Riebler

References

Rue H., Martino S, and Chopin N. (2009). Approximate Bayesian Inference for Latent Gaussian Models Using Integrated Nested Laplace Approximations. Journal of the Royal Statistical Society B ***71****: 319–392. (www.r-inla.org)*

Simpson DP, Martins TG, Riebler A, Fuglstad GA, Rue H, Sorbye SH (2014) Penalised Model Component Complexity: A principled, Practical Approach to Constructing Priors. Arxiv e-prints. 1403.4630

Guo, J., Riebler, A. and Rue H. (2017) Bayesian bivariate meta-analysis of diagnostic test studies with interpretable priors. Statistics in Medicine ***36(19)****: 3039–3058.*

See Also

`makeData`, `makePrior`, `runModel`, `forest`, `SROC`, `crosshair`.

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
  require("INLA", quietly = TRUE)
  data(Catheter)
 meta4diag(data = Catheter, model.type = 1, var.prior = "invgamma", cor.prior = "normal",
    var.par = c(0.25, 0.025), cor.par = c(0, 5), init = c(0.01, 0.01, 0),
   link = "logit", quantiles = c(0.025, 0.5, 0.975), verbose = FALSE, covariates = NULL,
    nsample = FALSE)
}

## End(Not run)
```

`meta4diagGUI` *A graphic user interface for meta4diag.*

Description

A graphic user interface for meta4diag which is fully interactive but still provides full functionality.

Usage

```
meta4diagGUI()
```

Author(s)

Jingyi Guo and Andrea Riebler

References

Havard Rue, Sara Martino, and Nicholas Chopin (2009). Approximate Bayesian Inference for Latent Gaussian Models Using Integrated Nested Laplace Approximations. Journal of the Royal Statistical Society B, 71, 319-392. (www.r-inla.org)

See Also

`meta4diag`

Examples

```
## Not run: meta4diagGUI()
```

`plot` *Default meta4diag plotting.*

Description

Takes a `meta4diag` object produced by `meta4diag()` and plot the posterior marginals (possibly with underlying priors) for hyperparameters and fixed effects.

Usage

```
  ## S3 method for class 'meta4diag'
plot(x, var.type="var1", add=FALSE, overlay.prior = TRUE,
      save = FALSE, width=5, height=5, ...)
```

Arguments

`x`	A `meta4diag` object.
`var.type`	Variable type that is of interest. Options are "var1", "var2", "rho" and names for fixed effects, which can be found after calling `summary()` function.
`add`	If add is TRUE, the plots are added to an existing plot, otherwise a new plot is created.
`overlay.prior`	Boolean to indicate whether the prior will be plotted overlay or not only for hyperparameters.
`save`	If save is TRUE, the plots are saved (pdf format) automatically in the working directory. save could also be a file name, i.e. `save="plot1.eps"`, a file with name plot1 and eps format will be saved in the working directory.
`width`	The width when used for saving the plot, unit of inches is used.
`height`	The height when used for saving the plot, unit of inches is used.
`...`	Arguments to be passed to methods, such as graphical parameters (see par) such as "main", "sub", "xlab", "ylab".

Details

Posterior marginal distribution possibly with underlying prior distribution is plotted.

Value

Besides plotting, the function returns an invisible NULL.

Author(s)

Jingyi Guo <`jingyi.guo@math.ntnu.no`> and Andrea Riebler <`andrea.riebler@math.ntnu.no`>

See Also

`meta4diag`

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
  require("INLA", quietly = TRUE)
  data(Catheter)
  res <- meta4diag(data = Catheter)
  plot(res, var.type="var1")
}

## End(Not run)
```

`print` *print meta4diag object.*

Description

Takes an `meta4diag` object which is obtained by function `meta4diag()` and print the model and data information such as model type used in the model, number of studies in the dataset and the name of covariates which are contained in the model. The variable type used in `plot.meta4diag` can be found here.

Usage

```
  ## S3 method for class 'meta4diag'
print(x, ...)
```

Arguments

`x`	A `meta4diag` object.
`...`	Further arguments passed to or from other methods.

Details

The basic model setting and dataset information are printed.

Value

The return value is invisible NULL.

Author(s)

Jingyi Guo <`jingyi.guo@math.ntnu.no`> and Andrea Riebler <`andrea.riebler@math.ntnu.no`>

See Also

`plot`

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
  require("INLA", quietly = TRUE)
  data(Catheter)
  res <- meta4diag(data = Catheter)
  print(res)
}

## End(Not run)
```

`runModel` *Run the bivariate model.*

Description

Run the bivariate model with INLA. This function is used within the main function `meta4diag()` and can also be used as a separate function.

Usage

```
runModel(outdata, outpriors, link = "logit",
  quantiles = c(0.025, 0.5, 0.975), verbose = FALSE, num.threads = 1)
```

Arguments

`outdata`	A data file for internal use.
`outpriors`	A list of prior settings prepared for internal use, see `makePriors`.
`link`	A string specifying the link function used in the model. Options are "logit", "probit" and "cloglog".

`quantiles`	A vector of quantiles, p(0), p(1),... to compute for each posterior marginal. The function returns, for each posterior marginal, the values x(0), x(1),... such that $Prob(X < x) = p.$ The default value are c(0.025, 0.5, 0.975). Not matter what other values are going to be given, the estimates for these 3 quantiles are always returned.
`verbose`	Boolean (default:FALSE) indicating whether the program should run in a verbose mode.
`num.threads`	Maximum number of threads the inla-program will use. xFor Windows this defaults to 1, otherwise its defaults to NULL (for which the system takes over control).

Value

A `INLA` object which will be used into function `makeObject()`.

Author(s)

Jingyi Guo and Andrea Riebler

References

Havard Rue, Sara Martino, and Nicholas Chopin (2009). Approximate Bayesian Inference for Latent Gaussian Models Using Integrated Nested Laplace Approximations. Journal of the Royal Statistical Society B, 71, 319-392. (www.r-inla.org)

See Also

`makeData`, `makePriors`, `makeObject`, `meta4diag`, `inla`

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)) {
   require("INLA", quietly = TRUE)
   data(Catheter)

   outdata = makeData(data=Catheter,model.type=1,covariates="type")
   outpriors = makePriors(var.prior = "invgamma", cor.prior = "normal",
                          var.par = c(0.25, 0.025), cor.par = c(0, 5))
   runModel(outdata, outpriors, link = "logit",
            quantiles = c(0.025, 0.5, 0.975), verbose = FALSE)
}

## End(Not run)
```

`Scheidler` *Scheidler dataset.*

Description

The dataset is used to compare the utility of three types of diagnostic imaging to detect lymph node metastases in patients with cervical cancer: lymphangiography (LAG), computed tomography (CT) and magnetic resonance (MR).

Usage

```
data(Scheidler)
```

Format

A data frame with 44 rows and 6 columns. Each row provides data for one study. The columns are "studynames" indicates the name of studies, "modality" indicates which diagnostic study that the observation is conducted from and is considered as a covariate, "TP", "FP", "FN" and "TN" are the number of true positive, false positive, false negative and true negative, respectively.

References

Scheidler J, Hricak H, Kyle K Y, et al. Radiological evaluation of lymph node metastases in patients with cervical cancer: a meta-analysis. Jama, 1997, 278(13): 1096-1101.

Examples

```
  data(Scheidler)
```

`SROC` *A standard summary receiver operating chracteristic plot.*

Description

Takes a `meta4diag` object produced by `meta4diag()` and makes the standard summary receiver operating chracteristic plot.

Usage

```
SROC(x,...)
## S3 method for class 'meta4diag'
SROC(x, sroc.type=1, est.type="mean", sp.cex=1.5,sp.pch=8,sp.col="red",
             dataShow="o", data.col="#FF0000", data.cex="scaled", data.pch=1,
             lineShow=T, line.lty=1, line.lwd=2, line.col="black",
             crShow=T, cr.lty=2, cr.lwd=1.5, cr.col="blue",
             prShow=T, pr.lty=3, pr.lwd=1,  pr.col="darkgray",
             dataFit = T, add=FALSE, main="", xlim, ylim,
             legend=F,legend.cex = 0.7,...)
```

Arguments

`x`	A `meta4diag` object.
`sroc.type`	A numerical value specifying the function used to make SROC line. Options are 1, 2, 3, 4, 5. When `sroc.type=1`, the SROC line is plotted as " The regression line 1" according to Arends et al.(2008), $$y = \mu + \rho\sqrt{\frac{\sigma_{\mu}^{2}}{\sigma_{\nu}^{2}}}(x-\nu)$$ When `sroc.type=2`, the SROC line is plotted as "The major axis method", $$y = \frac{\sigma_{\mu}^{2} - \sigma_{\nu}^{2} \pm \sqrt{(\sigma_{\mu}^{2} - \sigma_{\nu}^{2})^{2} + 4\rho^{2}\sigma_{\mu}^{2}\sigma_{\nu}^{2}}}{2\rho\sqrt{\sigma_{\mu}^{2}\sigma_{\nu}^{2}}}(x-\nu)+\mu$$ When `sroc.type=3`, the SROC line is plotted as "The Moses and Littenberg's regression line", $$y = \frac{\sigma_{\mu}^{2} + \rho\sqrt{\sigma_{\mu}^{2}\sigma_{\nu}^{2}}}{\sigma_{\nu}^{2} + \rho\sqrt{\sigma_{\mu}^{2}\sigma_{\nu}^{2}}}(x-\nu)+\mu$$ When `sroc.type=4`, the SROC line is plotted as "The regression line 2", $$y = \mu + \frac{1}{\rho}\sqrt{(}\frac{\sigma_{\mu}^{2}}{\sigma_{\nu}^{2}})(x-\nu)$$ When `sroc.type=5`, the SROC line is plotted as "The Rutter and Gatsonis's SROC curve", $$y = \mu + \sqrt{\frac{\sigma_{\mu}^{2}}{\sigma_{\nu}^{2}}}(x-\nu)$$
`est.type`	The estimates type used to make SROC plot. Options are "mean" and "median".
`sp.cex`	Summary points size. The summary points are mean or median of sensitivities and specificities of all sudies.
`sp.pch`	Point symbols of summary points.
`sp.col`	Color of summary points.
`dataShow`	A character indicating whether the original dataset or the fitted dataset is shown or not. If is "o", the original data will be plotted. If is "f", the fitted data will be plotted. No dataset will be plotted if `dataShow` is not "o" or "f".
`data.col`	Color of orignal data bubbles.
`data.cex`	A string or a numerical value indicating the size of the plotted dataset points. If is "bubble" or "scaled", the size of the data points are proportional to the total number of individuals in each study.
`data.pch`	A string or a numerical value indicating the symbol of the plotted dataset points.
`lineShow`	Boolean indicating whether the SROC line is shown or not.
`line.lty`	SROC line type.

`line.lwd`	SROC line width.
`line.col`	Color for the SROC line.
`crShow`	Boolean indicating whether the confidence region is shown or not.
`cr.lty`	Confidence region line width.
`cr.lwd`	Confidence region line width.
`cr.col`	Color for the confidence region line.
`prShow`	Boolean indicating whether the prediction region is shown or not.
`pr.lty`	Prediction region line type.
`pr.lwd`	Prediciton region line width.
`pr.col`	Color for the prediction region line.
`dataFit`	Boolean indicating the length SROC line. Either plotted from -1 to 1, or fit the data.
`add`	If add is TRUE, the plots are added to an existing plot, otherwise a new plot is created.
`main`	A overall title for the plot.
`xlim`	A numeric value, giving the x coordinates ranges.
`ylim`	A numeric value, giving the y coordinates ranges.
`legend`	Boolean indicating whether the legend is shown or not. Can also be a string indicating the position to show the legend. Options are "left", "bottom" and "right".
`legend.cex`	Lengend size.
`...`	Further arguments passed to or from other methods.

Details

The `SROC()` returns a plot of the true positive rate against the false positive rate (1-specificity) (ROC space) for the different possible cutpoints of a diagnostic test. It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity). The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test. The area under the curve is a measure of text accuracy. Summary points, summary receiver operating characteristic line, confidence region and prediction region of the summary points are plotted.

The function `SROC()` gives the user flexibility to control the plots.

Value

The return value is a list of the files created (if any).

Author(s)

Jingyi Guo and Andrea Riebler

References

Chappell F, Raab G, Wardlaw J (2009). "When are Summary ROC Curves Appropriate for Diagnostic Meta-analyses?" Statistics in Medicine, 28(21), 2653-2668.

Arends et al.(2008) "Bivariate random effects meta-analysis of ROC curves." Medical Decision Making, 28(5), 621-638.

See Also

`meta4diag`

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
  require("INLA", quietly = TRUE)
  data(Catheter)
  res <- meta4diag(data = Catheter)
  SROC(res, est.type="mean")
}

## End(Not run)
```

`summary` *Provide summary information.*

Description

Takes a fitted `meta4diag` object produced by `meta4diag()` and produces a summary from it.

Usage

```
  ## S3 method for class 'meta4diag'
summary(object,...)
```

Arguments

`object`	a fitted `meta4diag` object as produced by `meta4diag()`.
`...`	other arguments may be useful.

Details

Posterior mean and standard deviation (together with quantiles) are printed for the fixed effects in the model.

For the random effects the function `summary()` prints the posterior mean and standard deviations of the hyperparameters.

Value

`summary.meta4diag` returns a list with components:

`cpu.used`	The cpu time used to fit the corresponding model and data.
`summary.fixed`	The posterior mean and standard deviation (together with quantiles) for the fixed effects.
`summary.expected.g`	The posterior mean and standard deviation (together with quantiles) for the summarized fixed effects.
`summary.hyperpar`	The posterior mean and standard deviation (together with quantiles) for model hyperparameters.
`mlik`	The marginal log-likelihood of the model.
`var.type`	The variables type used in the `plot()` function.

Author(s)

Jingyi Guo `<jingyi.guo@math.ntnu.no>` and Andrea Riebler `<andrea.riebler@math.ntnu.no>`

See Also

`plot`,`meta4diag`

Examples

```
## Not run:
if(requireNamespace("INLA", quietly = TRUE)){
  require("INLA", quietly = TRUE)
  data(Catheter)
  res <- meta4diag(data = Catheter)
  summary(res)
  }

## End(Not run)
```

`table_cor` *The example table prior for correlation.*

Description

An example of a table prior density for the correlation parameter.

Usage

```
data(table_cor)
```

Format

A data frame with 2 columns named "x" and "y". The first column indicates the value of correlation points, usually in [-1,1]. The second column indicates the density of the corresponding points.

Examples

```
data(table_cor)
plot(table_cor)
```

`table_var` *The example table prior for variance.*

Description

An example of a table prior density for the variance parameter.

Usage

```
data(table_var)
```

Format

A data frame with 2 columns named "x" and "y". The first column indicates the value of variance points, usually in [0, some positive value]. The second column indicates the density of the corresponding points.

Examples

```
data(table_var)
plot(table_var)
```

`Telomerase` *Telomerase dataset.*

Description

The dataset contains 10 studies from a meta-analysis that uses the telomerase marker for the diagnosis of bladder cancer.

Usage

```
data(Telomerase)
```

Format

A data frame with 10 rows and 5 columns. Each row provides data for one study. The columns are "studynames" indicates the name of studies, "TP", "FP", "FN" and "TN" which are the number of true positive, false positive, false negative and true negative, respectively.

References

Glas AS, Roos D, Deutekom M, Zwinderman AH, Bossuyt PM, Kurth KH (2003). "Tumor Markers in the Diagnosis of Primary Bladder Cancer. A Systematic Review." The Journal of Urology, 169(6), 1975-1982.

Examples

```
data(Telomerase)
```

Index