

Package ‘BayesCombo’

September 20, 2016

Title Bayesian Evidence Combination

Version 1.0

Date 2016-09-20

Author Bruno Contrino, Stanley E. Lazic

Maintainer Stanley E. Lazic <stan.lazic@cantab.net>

Description Combining diverse evidence across multiple studies using Bayes factors.

Depends R (>= 2.10),

License GPL-3

LazyData true

Suggests knitr, rmarkdown, testthat (>= 0.7.1.99)

URL <https://github.com/stanlazic/BayesCombo>

BugReports <https://github.com/stanlazic/BayesCombo/issues>

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

BSfactor	2
calc.post.beta	3
forestplot	3
GEdata	4
plot.PMP	5
plot.PMPlist	6
pmp	7
pmp.update	8
summary.PMP	10
Index	11

BSfactor

*Calculate a 'Bayesian Safety' (BS) factor***Description**

Calculates the prior probability for the null that gives a just significant result.

Usage

```
BSfactor(x, n = 100, sig = 0.95)
```

Arguments

x	An object of class PMPList.
n	Number of steps between 0.33333 and 0.99999 to calculate the BS factor. Increasing n gives a more accurate estimate.
sig	The 'significance' cut-off value for a non-null posterior model probability.

Details

When the posterior probability for a non-null hypothesis is large, it may be of interest to calculate the prior for the null that gives a just significant result. Here 'significant' is a threshold that the posterior must exceed. For example, if a series of experiments on homeopathy gives a posterior probability of 0.98 in favour homeopathy's effectiveness, we can calculate the strength of the prior needed to make the posterior just pass the 0.98 threshold. If the BS factor is very large (sceptical prior), then we may be more inclined to believe the results because they were sufficient to shift a strong prior belief in no effect. If the data can only overcome a weak prior, then the results may be unconvincing, despite having a large posterior probability. We can ask ourselves if our prior is greater or less than the BS factor, and judge the results accordingly.

Value

Object of class 'BSfactor' which contains a matrix of final posterior model probabilities, a matrix of prior model probabilities, the 'boundary' and which hypothesis is being considered.

See Also

[pmp.update](#)

Examples

```
x <- pmp.update(beta = c(0.126, 5.005, 1.298, 0.000476),
  se.beta = c(0.0504, 2.5811, 2.0541, 0.0026) )
```

```
BSfactor(x)
```

calc.post.beta	<i>Utility functions</i>
----------------	--------------------------

Description

Not called directly by user.

Usage

```
calc.post.beta(beta, se.beta, beta0, se0)
```

```
calc.post.se(se.beta, se0)
```

```
prior.se(beta, se.beta, percent)
```

Arguments

beta	An numeric effect size.
se.beta	A numeric standard error of the effect.
beta0	A numeric prior value for the effect size.
se0	A numeric prior variance.
percent	A numeric value that is used to calculate the prior variance.

forestplot	<i>Forest plot</i>
------------	--------------------

Description

Plots the effect sizes and standard errors for a set of studies as a standard forest plot. Also shown are the priors for each study.

Usage

```
forestplot(x, range = NULL, xlab = "Effect size", ylab = "Study", ...)
```

Arguments

x	An object of the class PMPlist.
range	A numeric vector of length two that sets the range of the x-axis. Values are passed to xlim. Sensible values are used default when range=NULL.
xlab, ylab	Text for the x and y axes labels.
...	Other options passed to plot().

Details

This function plots the data from the `pmp.update` function. It may be hard to visualise the results if the effect sizes differ. In this case it may be preferable to standardise the effect sizes by setting `scale=TRUE` in the `pmp.update` function.

Value

A forest plot.

See Also

[pmp.update](#)

Examples

```
x <- pmp.update( beta = c(0.0126, 5.0052, 1.2976, 0.0005),
  se.beta = c(0.050, 2.581, 2.054, 0.003) )
forestplot(x)
```

GEdata

Gene expression data set

Description

Effect sizes and standard errors for an Affymetrix microarray experiment. Only subset of the full experiment is used and the results are for two experimental groups in two different tissues.

Format

A data frame with 25206 rows of gene probes and 7 columns:

probeset: Affymetrix probe set ID.

beta.sk: The effect size from the gene expression output for the skeletal muscle tissue.

beta.ab: The effect size from the gene expression output for the abdominal fat tissue.

se.sk: The standard error from the gene expression output for the skeletal muscle tissue.

se.ab: The standard error from the gene expression output for the abdominal fat tissue.

Aexprs.sk: Average expression in skeletal muscle.

Aexprs.ab: Average expression in abdominal fat.

Details

The study design has 4 sets of 7 mice, a sham group, a TSC22D4-knockout group, a LCN13-knockout and a TSC22D4-knockout LCN13-knockout group. Only the sham and the TSC22D4-knockout group are used in this data set.

References

Friedrich K, Jones A, Seibert O, Sijmonsma TP, Wang X, Sticht C, Gretz N, Fleming T, Nawroth PP, Stremmel W, Rose AJ, Diaz MB, Bluher M, Herzig (2014). Transforming growth factor beta-like stimulated clone 22 D4 promotes diabetic hyperglycemia and insulin resistance. *GEO ID: GSE53754*.

plot.PMP

Plot of prior, data, and posterior distributions

Description

Plots the results from a single study.

Usage

```
## S3 method for class 'PMP'
plot(x, range = NULL, n = 200, leg.loc = "topleft",
     xlab = "Effect size", ylab = "", ...)
```

Arguments

x	A PMP object created by the pmp() function.
range	A numeric vector of length two that sets the range of the x-axis. Values are passed to xlim. Sensible values are used default when range=NULL.
n	An integer that specifies the number of x-values use when plotting the distributions.
leg.loc	Location of the legend; default is top left. See the legend help pages for all the options. If leg.loc = NULL the legend is not plotted.
xlab, ylab	Text for the x and y axes labels.
...	Other options passed to plot().

Details

Plots the prior, data (likelihood), and posterior distribution calculated from the pmp() function for a single study.

Value

Plot of distributions.

See Also

[pmp](#)

Examples

```
x <- pmp(beta = 5.005, se.beta = 2.05)
plot(x)
```

plot.PMPlist	<i>Plot of posterior model probabilities</i>
--------------	--

Description

Plots the updated posterior model probabilities as each study is added.

Usage

```
## S3 method for class 'PMPlist'
plot(x, leg.loc = "topleft", lty = 1, ...)
```

Arguments

x	An object of the class PMPlist created by the pmp.update() function.
leg.loc	Location of the legend; default is top left. See the legend help pages for all the options.
lty	An vector of integers that specifies the line types. See the matplot() function for details.
...	Other options passed to matplot().

Details

Each line is a hypothesis and they start at the prior model probabilities (usually 1/3).

Value

Plot of posterior model probabilities.

See Also

[pmp.update](#)

Examples

```
x <- pmp.update(beta = c(0.0126, 5.0052, 1.2976, 0.0005),
  se.beta = c(0.050, 2.581, 2.054, 0.003) )
plot(x)
```

pmp

*Calculates posterior model probabilities for one study***Description**

The function takes a single effect size and a its standard error and calculates the posterior model probabilities for each hypothesis (H<: the effect size is less than 0, H0: the effect size is zero, or H>: the effect size is greater than zero).

Usage

```
pmp(beta, se.beta, beta0 = 0, percent = 99, var.mult = 1, H0 = c(0, 0),
    mod.priors = rep(1/3, 3), scale = FALSE, adjust = FALSE,
    epsilon = 1e-06, adj.factor = 1e-04)
```

Arguments

beta	An numeric effect size.
se.beta	A numeric standard error of the effect.
beta0	A numeric prior value for the effect size. Default is zero.
percent	A numeric value that is used to calculate the prior variance. The default value of 99 calculates the prior variance so that the 99 from zero) confidence intervals of the data distribution.
var.mult	Variance multiplier used to increase or decrease the prior variance.
H0	A vector of length two that defines the null hypothesis. If the values are identical (e.g. $H_0 = c(0, 0)$) a point null is used, otherwise the null is defined as the range between the lower and upper value.
mod.priors	Prior model probabilities; default is an equal probability of 1/3.
scale	Logical. Whether to scale the effect size by its standard error. This has no effect on the calculations but it may be useful to use standardised effect sizes.
adjust	Logical, whether to adjust the posterior model probabilities if they are too small. When combining multiple studies, this prevents a single study from having too much influence on the results. For example, if the probability for a hypothesis is zero, then additional studies cannot alter this probability.
epsilon	A small value that a posterior model probability must be below before an adjustment is made.
adj.factor	A small number added to each posterior model probability if adjust=TRUE and a PMP is less than epsilon. The PMPs are then re-scaled to sum to one.

Details

Effect sizes are typically a mean difference between groups, a regression slope, odds ratio, and so on, that are provided by generalised linear models. The standard errors for the effect size are provided in the output from functions that calculate the effect size. Two types of priors need to be specified. The first is the prior for the effect size, which is given by a mean (usually zero) and variance, which is calculated automatically. Second, prior probabilities for each hypothesis need to be provided, and an equal probability of 1/3 is used by default.

The null hypothesis (usually effect size = 0) can be specified as either a point null or as a range. A point null tests if effect size is *exactly* zero while the range null tests if the effect size close enough to zero for practical purposes. "Close enough" is defined by the user as a range on either size of zero.

To combine multiple effect sizes use the `pmp.update` function.

Value

Object of class `pmp` which contains the posterior model probabilities and other calculate values.

See Also

[plot.PMP](#), [pmp.update](#)

Examples

```
# library(labstats) # need to install separately
# plot(time.immob ~ dose, data=fluoxetine)
# summary(lm(time.immob ~ dose, data=fluoxetine))
x <- pmp(beta=-0.25200, se.beta=0.09913) # dose effect from above output
x
```

`pmp.update`

Calculates posterior model probabilities for multiple studies

Description

The function takes multiple effect size and a their standard errors and calculates the posterior model probabilities for each hypothesis (H<: the effect size is less than 0, H0: the effect size is zero, or H>: the effect size is greater than zero).

Usage

```
pmp.update(beta, se.beta, beta0 = 0, percent = 99, H0 = c(0, 0),
  scale = FALSE, mod.priors = rep(1/3, 3), var.mult = 1, adjust = FALSE,
  epsilon = 1e-06, adj.factor = 1e-04, ...)
```


Arguments

beta	An numeric effect size.
se.beta	A numeric standard error of the effect.
beta0	A numeric prior value for the effect size. Default is zero.
percent	A numeric value that is used to calculate the prior variance. The default value of 99 calculates the prior variance so that the 99 from zero) confidence intervals of the data distribution.
H0	A vector of length two that defines the null hypothesis. If the values are identical (e.g. $H_0 = c(0, 0)$) a point null is used, otherwise the null is defined as the range between the lower and upper value.
scale	Logical. Whether to scale the effect size by its standard error. This has no effect on the calculations but it may be useful to use standardised effect sizes.
mod.priors	Prior model probabilities; default is an equal probability of 1/3.
var.mult	Variance multiplier used to increase or decrease the prior variance.
adjust	Logical, whether to adjust the posterior model probabilities if they are too small. When combining multiple studies, this prevents a single study from having too much influence on the results. For example, if the probability for a hypothesis is zero, then additional studies cannot alter this probability.
epsilon	A small value that a posterior model probability must be below before an adjustment is made.
adj.factor	A small number added to each posterior model probability if adjust=TRUE and a PMP is less than epsilon. The PMPs are then re-scaled to sum to one.
...	Options to be passed to pmp .

Details

This function calls [pmp](#) once for each study to be combined where the posterior model probabilities for one study are used as the priors for the next study. The inputs are identical to the [pmp](#) function, so please see that documentation for details.

Value

Object of class `PMPList` which contains a matrix of posterior model probabilities for each updated step and other calculate values

Examples

```
x <- pmp.update(beta = c(0.0126, 5.0052, 1.2976, 0.0005),
  se.beta = c(0.050, 2.581, 2.054, 0.003) )
x
plot(x)
```

`summary.PMP`*Print and summary for PMP, PMPlist, and BSfactor objects*

Description

Print and summary methods for PMP, PMPlist,BSfactor objects.

Usage

```
## S3 method for class 'PMP'
summary(x, digits = 4)

## S3 method for class 'PMP'
print(x, digits = 4)

## S3 method for class 'PMPlist'
summary(x, digits = 4)

## S3 method for class 'PMPlist'
print(x, digits = 4)

## S3 method for class 'BSfactor'
summary(x, digits = 4)
```

Arguments

<code>x</code>	A PMP, PMPlist, or BSfactor object.
<code>digits</code>	Number of digits to show.

Value

A list of posterior model probabilities and calculated values.

Index

BSfactor, [2](#)

calc.post.beta, [3](#)

calc.post.se (calc.post.beta), [3](#)

forestplot, [3](#)

GEdata, [4](#)

plot.PMP, [5](#), [8](#)

plot.PMPlist, [6](#)

pmp, [5](#), [7](#), [9](#)

pmp.update, [2](#), [4](#), [6](#), [8](#), [8](#)

print.PMP (summary.PMP), [10](#)

print.PMPlist (summary.PMP), [10](#)

prior.se (calc.post.beta), [3](#)

summary.BSfactor (summary.PMP), [10](#)

summary.PMP, [10](#)

summary.PMPlist (summary.PMP), [10](#)