Package ‘MetaStan’

April 9, 2020

Version 0.2.0
Title Bayesian Meta-Analysis via ‘Stan’
Description Performs Bayesian meta-analysis and model-based meta-analysis using ‘Stan’. Includes binomial-normal hierarchical models and option to use weakly informative priors for the heterogeneity parameter and the treatment effect parameter which are described in Guenhan, Roever, and Friede (2020) <doi:10.1002/jrsm.1370>.
License GPL (>= 3)
Encoding UTF-8
Date 2020-04-06
LazyData true
ByteCompile true
Depends R (>= 3.4.0), Rcpp (>= 0.12.17), methods
Imports rstan (>= 2.18.1), rstantools (>= 1.5.0)
Suggests testthat, knitr, rmarkdown, ggplot2
LinkingTo StanHeaders (>= 2.18.0), rstan (>= 2.18.1), BH (>= 1.66.0-1), Rcpp (>= 0.12.17), RcppEigen (>= 0.3.3.4.0)
SystemRequirements GNU make
NeedsCompilation yes
RoxygenNote 7.1.0
URL https://github.com/gunhanb/MetaStan
BugReports https://github.com/gunhanb/MetaStan/issues
VignetteBuilder knitr
Author Burak Kuersad Guenhan [aut, cre]
      (<https://orcid.org/0000-0002-7454-8680>),
Maintainer Burak Kuersad Guenhan <burak.gunhan@med.uni-goettingen.de>
Repository CRAN
Date/Publication 2020-04-09 13:20:09 UTC
Description

Fitting Bayesian meta-analysis models via Rstan.

Details

To fit meta-analysis models using frequentist methods, there are many R packages available including ‘metafor’. On the other hand, Bayesian estimation methods such as Markov chain Monte Carlo (MCMC) are very attractive for meta-analysis, especially because they can be used to fit more complicated models. These include binomial-normal hierarchical models and beta-binomial models which are based on the exact distributional assumptions unlike (commonly used) normal-normal hierarchical model. Another advantage of Bayesian methods to be able to use informative prior distributions for example to regularize heterogeneity estimates in case of low number of studies. Thus, we developed ‘MetaStan’ which uses Stan (a modern MCMC engine) to fit several pairwise meta-analysis models including binomial-normal hierarchical model and beta-binomial model. This package is also the accompanying package of Guenhan et al (2018).

Author(s)

Burak Kuersad Guenhan <burak.guenhan@med.uni-goettingen.de>

References


**convert_data_arm**

*Convert contrast-based dataset to arm-based dataset*

**Description**

`convert_data_arm` creates a dataframe to fit a Binomial-Normal Hierarchical model using `glmer` function.

**Usage**

`convert_data_arm(nt, nc, pt, pc)`

**Arguments**

- `nt`: Number of subjects in treatment arm
- `nc`: Number of subjects in control arm
- `pt`: Number of events in treatment arm
- `pc`: Number of events in treatment arm

**Value**

A dataframe object

**Examples**

```r
data('dat.Crins2014', package = "MetaStan")
## Subset of dataset where PTLD outcomes available
dat.Crins2014.PTLD = subset(dat.Crins2014, is.finite(exp.PTLD.events))
## Create arm-based dataset
dat.Crins2014.PTLD.arm <- convert_data_arm(dat.Crins2014.PTLD$exp.total,
                                         dat.Crins2014.PTLD$cont.total,
                                         dat.Crins2014.PTLD$exp.PTLD.events,
                                         dat.Crins2014.PTLD$cont.PTLD.events)
```

**create_MBMA_dat**

*Prepare model-based meta-analysis dataset for Stan.*

**Description**

`create_MBMA_dat` converts datasets in the one-study-per-row format to one-arm-per-row format,
Usage

create_MBMA_dat(
  dat = dat,
  armVars = c(dose = "t", responders = "r", sampleSize = "n"),
  nArmsVar = "nd"
)

Arguments

dat Data in one-study-per-row format.
armVars Vector of per-arm variables The name of each component will be the column name in the resulting dataset.
nArmsVar Variable holding the number of arms for each study.

Details

The resulting data.frame can be used as data argument in MBMA_stan.

Value

A data frame with the generated columns.

Author(s)

Burak Kuersad Guenhan, <burak.gunhan@med.uni-goettingen.de> and Gert van Valkenhoef

See Also

gemtc::mtc.data.studyrow and nmaINLA::create_INLA_dat

Examples

data('dat.Eletriptan')
## Create the dataset suitable for MBMA_stan
EletriptanDat <- create_MBMA_dat(dat = dat.Eletriptan,
  armVars = c("dose" = "d", "r" = "r", "n" = "n"), nArmsVar = 'nd')
## Check that the data are correct
print(EletriptanDat)
**dat.Berkey1995**

Trials investigating effectiveness of the BCG vaccine against TB

**Description**

A dataset containing the results from 13 trials examining the efficacy of Bacillus Calmette-Guerin (BCG) vaccine against tuberculosis (TB).

**Usage**

dat.Berkey1995

**Format**

A data frame with following columns

<table>
<thead>
<tr>
<th>Trial</th>
<th>Trial number</th>
</tr>
</thead>
<tbody>
<tr>
<td>rt</td>
<td>number of TB events in treatment arm</td>
</tr>
<tr>
<td>nt</td>
<td>number of subjects in treatment arm</td>
</tr>
<tr>
<td>rc</td>
<td>number of TB events in control arm</td>
</tr>
<tr>
<td>nc</td>
<td>number of subjects in control arm</td>
</tr>
</tbody>
</table>

**Latitude** absolute latitude of the study location

**Source**


---

**dat.Crins2014**

Pediatric liver transplant example data

**Description**

Numbers of cases and events (PTLDs or deaths) in experimental and control groups of six studies.

**Usage**

dat.Crins2014
Format

A data frame with following columns

- **publication** publication identifier (first author and publication year)
- **year** publication year
- **randomized** randomization status (y/n)
- **control.type** type of control group ("concurrent" or "historical")
- **comparison** type of comparison ("IL-2RA only", "delayed CNI", or "no/low steroids")
- **followup** t follow-up time in months
- **exp.PTLD.events** number of PTLD events in experimental group
- **exp.death.events** number of deaths in experimental group
- **exp.total** number of patients in experimental group
- **cont.PTLD.events** number of PTLD events in control group
- **cont.death.events** number of deaths in control group
- **cont.total** number of patients in control group

Source


---

**dat.Eletriptan**  
*Migraine pain relief example (Eletriptan)*

Description

Numbers of patients and events (headache free at 2 hours) in experimental and control groups of 12 studies. It is in one-study-per-row format.

Usage

dat.Eletriptan

Format

A data frame with following columns

- **ID** trial ID
- **d1** dose (mg) in the first arm (placebo)
- **r1** number of events in the first arm (placebo)
- **n1** number of patients in the first arm (placebo)
- **d2** dose (mg) in the second arm
MBMA_stan

\[ r_2 \] number of events in the second arm
\[ n_2 \] number of patients in the second arm
\[ d_3 \] dose (mg) in the third arm
\[ r_3 \] number of events in the third arm
\[ n_3 \] number of patients in the third arm
\[ d_4 \] dose (mg) in the fourth arm
\[ r_4 \] number of events in the fourth arm
\[ n_4 \] number of patients in the fourth arm
\[ nd \] number of treatment arms

Source

---

Fitting a model-based meta-analysis model using Stan

Description
‘MBMA_stan’ fits a model-based meta-analysis model using Stan.

Usage

```r
MBMA_stan(
  data = NULL,
  Pred_doses = NULL,
  model = "AB_Emax",
  mu_prior = c(0, 10),
  Emax_prior = c(0, 10),
  ED50_prior = c(0, 10),
  tau_prior = 0.5,
  tau_prior_dist = "half-normal",
  chains = 4,
  iter = 2000,
  warmup = 1000,
  adapt_delta = 0.95
)
```
Arguments

data
An object of ‘create_MBMA_dat’.

Pred_doses
A numerical vector specifying the doses which prediction will be made. Default is NULL.

model
A string specifying the model used. Available options are ‘Baseline_Emax’ (Baseline random effects model (Boucher and Bennets, 2016)), ‘CB_Emax’ (contrast-based random effects model (Mawdsley et al., 2016)), ‘AB_Emax’ (arm-based random effects model, adapted from (Zhang et al., 2017)), ‘CBPlusBaseline_Emax’ (contrast-based plus baseline random effects model, adapted from (Dias et al., 2013)). Default is ‘AB_Emax’.

mu_prior
A numerical vector specifying the parameter of the normal prior density for baseline risks, first value is parameter for mean, second is for variance. Default is c(0, 10).

Emax_prior
A numerical vector specifying the parameter of the normal prior density for Emax parameter, first value is parameter for mean, second is for standard deviation. Default is c(0, 10).

ED50_prior
A numerical vector specifying the parameter of the normal prior density for ED50 parameter, first value is parameter for mean, second is for standard deviation. Default is c(0, 10).

tau_prior
A numerical value specifying the standard deviation of the prior density for heterogeneity stdev. Default is 0.5.

tau_prior_dist
A string specifying the prior density for the heterogeneity standard deviation, option is ‘half-normal’ for half-normal prior, ‘uniform’ for uniform prior, ‘half-cauchy’ for half-cauchy prior.

chains
A positive integer specifying the number of Markov chains. The default is 4.

iter
A positive integer specifying the number of iterations for each chain (including warmup). The default is 2000.

warmup
A positive integer specifying the number of warmup (aka burnin) iterations per chain. The default is 1000.

adapt_delta
A numerical value specifying the target average proposal acceptance probability for adaptation. See Stan manual for details. Default is 0.95. In general you should not need to change adapt_delta unless you see a warning message about divergent transitions, in which case you can increase adapt_delta from the default to a value closer to 1 (e.g. from 0.95 to 0.99, or from 0.99 to 0.999, etc).

Value

an object of class ‘stanfit’ returned by ‘rstan::sampling’

References


Examples

```r
## Not run:
## Load the dataset
data('dat.Eletriptan', package = "MetaStan")
## Fitting a Binomial-Normal Hierarchial model using WIP priors
datMBMA = create_MBMA_dat(dat = dat.Eletriptan,
    armVars = c(dose = "d", responders = "r",
    sampleSize = "n"),
    nArmsVar = "nd")

MBMA.AB.Emax <- MBMA_stan(data = datMBMA,
    model = "AB_Emax",
    Pred_doses = seq(0, 80, length.out = 11),
    Emax_prior = c(0, 10),
    tau_prior_dist = "half-normal",
    tau_prior = 0.5)
```

## End(Not run)

### Description

`meta_stan` fits a meta-analysis model using Stan.

### Usage

```r
meta_stan(
    ntrt, nctrl, rtrt, rctrl, data = NULL,
    model = "BNHM1", mu_prior = c(0, 10),
    theta_prior = NULL, tau_prior = 0.5,
    tau_prior_dist = "half-normal", delta = NULL,
    chains = 4,
```
iter = 2000,
warmup = 1000,
adapt_delta = 0.95
)

Arguments

ntrt Number of subjects in treatment arm
nctrl Number of subjects in control arm
rtrt Number of events in treatment arm
rctrl Number of events in control arm
data Optional data frame containing the variables given to the arguments above
model A string specifying the model used. Available options are ‘FE’ (fixed-effect model using binomial likelihood) ‘BNHM1’ (Model 4 from Jackson et al (2018)), ‘BNHM2’ (Model 2 from Jackson et al (2018)), and ‘Beta-binomial’ (Beta-binomial model from Kuss (2014)). Default is ‘BNHM1’.
mu_prior A numerical vector specifying the parameter of the normal prior density for baseline risks, first value is parameter for mean, second is for variance. Default is c(0, 10).
theta_prior A numerical vector specifying the parameter of the normal prior density for treatment effect estimate, first value is parameter for mean, second is for variance. Default is NULL.
tau_prior A numerical value specifying the standard dev. of the prior density for heterogeneity stddev. Default is 0.5.
tau_prior_dist A string specifying the prior density for the heterogeneity standard deviation, option is ‘half-normal’ for half-normal prior, ‘uniform’ for uniform prior, ‘half-cauchy’ for half-cauchy prior.
delta A numerical value specifying the upper bound of the a priori interval for treatment effect on odds ratio scale Guenhan et al (2018). This is used to calculate a normal weakly informative prior for theta. Thus when this argument is specified, ‘theta’ should be left empty. Default is NULL.
chains A positive integer specifying the number of Markov chains. The default is 4.
iter A positive integer specifying the number of iterations for each chain (including warmup). The default is 2000.
warmup A positive integer specifying the number of warmup (aka burnin) iterations per chain. The default is 1000.
adapt_delta A numerical value specifying the target average proposal acceptance probability for adaptation. See Stan manual for details. Default is 0.95. In general you should not need to change adapt_delta unless you see a warning message about divergent transitions, in which case you can increase adapt_delta from the default to a value closer to 1 (e.g. from 0.95 to 0.99, or from 0.99 to 0.999, etc).

Value

an object of class ‘stanfit’ returned by ‘rstan::sampling’
References


Examples

```r
## Not run:
data('dat.Crins2014', package = "MetaStan")
## Subset of dataset where PTLD outcomes available
dat.Crins2014.PTLD = subset(dat.Crins2014, is.finite(exp.PTLD.events))
## Fitting a Binomial-Normal Hierarchical model using vague priors for theta
bnhm.vague.PTLD.stan <- meta_stan(ntrt = dat.Crins2014.PTLD$exp.total,
                                   nctrl = dat.Crins2014.PTLD$cont.total,
                                   rtrt = dat.Crins2014.PTLD$exp.PTLD.events,
                                   rctrl = dat.Crins2014.PTLD$cont.PTLD.event,
                                   mu_prior = c(0, 10), theta_prior = c(0, 100),
                                   tau_prior = 0.5)

## Obtaining a small summary
print(bnhm.vague.PTLD.stan)
## Extract the rstan fit for post-processing, eg convergence diagnostics
bnhm.vague.PTLD.stanfit = bnhm.vague.PTLD.stan$fit
## see `?rstan::sampling` for for post-processing of the `stanfit` object
## Fitting a Binomial-Normal Hierarchical model using WIP for theta
bnhm.wip.PTLD.stan <- meta_stan(ntrt = dat.Crins2014.PTLD$exp.total,
                                 nctrl = dat.Crins2014.PTLD$cont.total,
                                 rtrt = dat.Crins2014.PTLD$exp.PTLD.events,
                                 rctrl = dat.Crins2014.PTLD$cont.PTLD.event,
                                 mu_prior = c(0, 10),
                                 theta_prior = c(0, 2.82),
                                 tau_prior = 0.5)

## Fitting a fixed-effect Binomial model using vague priors for theta
bm.vague.PTLD.stan <- meta_stan(ntrt = dat.Crins2014.PTLD$exp.total,
                                 nctrl = dat.Crins2014.PTLD$cont.total,
                                 rtrt = dat.Crins2014.PTLD$exp.PTLD.events,
                                 rctrl = dat.Crins2014.PTLD$cont.PTLD.event,
                                 mu_prior = c(0, 10), theta_prior = c(0, 100),
                                 model = "FE")

## Fitting a Beta-binomial model using vague priors
bnhm.wip.PTLD.stan <- meta_stan(ntrt = dat.Crins2014.PTLD$exp.total,
                                 nctrl = dat.Crins2014.PTLD$cont.total,
                                 rtrt = dat.Crins2014.PTLD$exp.PTLD.events,
                                 rctrl = dat.Crins2014.PTLD$cont.PTLD.event,
                                 model = "Beta-binomial")

## End(Not run)
```
print.meta_stan

---

**print.meta_stan**  *Print meta_stan object*

---

**Description**

Takes an `meta_stan` object which is obtained by function `meta_stan` and print the model and data information such as model type used in the model.

**Usage**

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

**Arguments**

- `x`  
  A `meta_stan` object.

- `digits`  
  An integer indicating the number of decimal places.

- `...`  
  Further arguments passed to or from other methods.

**Details**

The resulting data.frame can be used as data argument in `meta_stan`.

**Value**

The return value is invisible NULL.
Index

*Topic **datasets**
  dat.Berkey1995, 5
  dat.Crins2014, 5
  dat.Eletriptan, 6

convert_data_arm, 3
create_MBMA_dat, 3

dat.Berkey1995, 5
dat.Crins2014, 5
dat.Eletriptan, 6

MBMA_stan, 7
meta_stan, 9
MetaStan (MetaStan-package), 2
MetaStan-package, 2

print.meta_stan, 12