Package 'breathteststan'

November 5, 2017

Type Package

Title Stan-Based Fit to Gastric Emptying Curves

Version 0.4.0

Description Stan-based curve-fitting function
for use with package 'breathtestcore' by the same author.
Stan functions are refactored here for easier testing.

License GPL-3

Depends R (>= 3.4.0)

Imports Rcpp, methods, tibble, purrr, dplyr, rstan(>= 2.16), stringr,
tidyr, breathtestcore(>= 0.4.0)

Suggests ggplot2, shinystan, bayesplot, testthat, covr, knitr,
rmarkdown

LinkingTo StanHeaders(>= 2.16.0.0), rstan(>= 2.16.0), BH(>= 1.62.0.1),
Rcpp(>= 0.12.0), RcppEigen(>= 0.3.3.0.0)

URL https://github.com/dmenne/breathteststan

BugReports https://github.com/dmenne/breathteststan/issues

NeedsCompilation yes

RoxygenNote 6.0.1

SystemRequirements pandoc

Author Dieter Menne [aut, cre],
Menne Biomed Consulting Tuebingen [cph],
Benjamin Misselwitz [fnd],
Mark Fox [fnd],
University Hospital of Zurich, Dep. Gastroenterology [fnd, dtc]

Maintainer Dieter Menne <dieter.menne@menne-biomed.de>

Repository CRAN

Date/Publication 2017-11-05 09:36:54 UTC

R topics documented:

`as.matrix.coef_diff_by_group_stan` *S3 as.matrix for result of coef_diff_by_group*

Description

Generates a matrix that can be used with plotting functions from package `mcmc_hist`.

Usage

```
## S3 method for class 'coef_diff_by_group_stan'
as.matrix(x, ...)
```

Arguments

`x`	Result of a call to `coef_diff_by_group(fit)`
`...`	parameter name as string, e.g. `"m"`, `"k"`, `"beta"`, `"t50_bluck_coward"`. When missing, `"t50_maes_ghoos"` is assumed.

Value

mcmc array with columns of differences for use with functions from packages bayesplot or coda

Examples

```
library(dplyr)
library(breathtestcore)
library(ggplot2)
data("usz_13c", package = "breathtestcore")
data = usz_13c %>%
  dplyr::filter( patient_id %in%  c("norm_001", "norm_002", "norm_003",
                        "norm_004", "pat_001", "pat_002","pat_003")) %>%
  cleanup_data()
fit = stan_group_fit(data, iter = 300, chains = 1)
cf = coef_diff_by_group(fit)
str(cf)
# Calling without parameters gives Maes/Ghoos t50
bayesplot::mcmc_hist(as.matrix(cf))
```

```
# Use a function from the bayesplot universe
dens = bayesplot::mcmc_dens(as.matrix(cf, parameter = "m"))
# use suppressMessages to avoid a message "another scale"
suppressMessages(
  dens + geom_vline(xintercept = 0) + scale_x_continuous(limits= c(-20,10)))
```

`coef_diff_by_group` *Tabulates breath test parameter differences of groups from Stan group fit*

Description

Given a Stan fit with grouping to 13C breath test curves, computes point estimated and Bayesian credible intervals for all group pair differences, for examples of the half emptying time `t50`.

Usage

```
## S3 method for class 'breathteststangroupfit'
coef_diff_by_group(fit, mcp_group = NULL,
  reference_group = NULL, ...)
```

Arguments

`fit`	Object of class `breathteststangroupfit` from `stan_fit`
`mcp_group`	Not used, always all pairs are compared
`reference_group`	Not used
`...`	Not used

Value

A `tibble` of class `coef_diff_by_group_stan` with columns

parameter Parameter of fit, e.g. `beta, k, m, t50`

method Method used to compute parameter. `exp_beta` refers to primary fit parameters `beta, k, m`.

groups Which pairwise difference, e.g `solid - liquid`

estimate Point estimate (chain mean) of the difference

cred.low, cred.high Lower and upper 95 percent credible interval of difference.

The chains of pairwise differences are returned as a attribute `chain` for use in plotting. See example below how to use these to display difference histograms.

Examples

```
library(dplyr)
library(breathtestcore)
data("usz_13c", package = "breathtestcore")
data = usz_13c %>%
  dplyr::filter( patient_id %in%
   c("norm_001", "norm_002", "norm_003", "norm_004", "pat_001", "pat_002","pat_003")) %>%
  cleanup_data()
fit = stan_group_fit(data, iter = 300, chains = 1) # Use more iterations!
cf = coef_diff_by_group(fit)
cc = attr(cf, "chain") %>%
   filter(key == "t50_maes_ghoos", abs(diff) < 200) %>%
   mutate(
     groups = paste(group2, group1, sep = " - ")
   )
str(cc)
if (require(ggplot2)) {
  ggplot(cc, aes(x = diff)) + geom_histogram() + facet_wrap(~groups)
}
# For comparison
fit = nlme_fit(data)
coef_diff_by_group(fit)
```

---

sigma.breathteststanfit

S3 method to exctract the residual standard deviation

Description

Functions for S3 method defined in breathtestcore for `stan_fit` and `stan_group fit`.

Usage

```
## S3 method for class 'breathteststanfit'
sigma(object, ...)
```

Arguments

`object`	A Stan-based fit
`...`	Not used

Value

A numeric value giving the sigma (= average residual standard deviation) term from the Stan fit.

stan_fit *Bayesian Stan fit to 13C Breath Data*

Description

Fits exponential beta curves to 13C breath test series data using Bayesian Stan methods. See https://menne-biomed.de/blog/breath-test-stan for a comparision between single curve, mixed-model population and Bayesian methods.

Usage

```
stan_fit(data, dose = 100, sample_minutes = 15, student_t_df = 10,
  chains = 2, iter = 1000, model = "breath_test_1")
```

Arguments

data	Data frame or tibble as created by cleanup_data, with mandatory columns patient_id, group, minute and pdr. It is recommended to run all data through cleanup_data which will insert dummy columns for patient_id and minute if the data are distinct, and report an error if not. Since the Bayesian method is stabilized by priors, it is possible to fit single curves.
dose	Dose of acetate or octanoate. Currently, only one common dose for all records is supported.
sample_minutes	If mean sampling interval is < sampleMinutes, data are subsampled using a spline algorithm
student_t_df	When student_t_df < 10, the student distribution is used to model the residuals. Recommended values to model typical outliers are from 3 to 6. When student_t_df >= 10, the normal distribution is used.
chains	Number of chains for Stan
iter	Number of iterations for each Stan chain
model	Name of model; use names(stanmodels) for other models.

Value

A list of classes "breathteststanfit" and "breathtestfit" with elements

- coef Estimated parameters as data frame in a key-value format with columns patient_id, group, parameter, metho and value. Has an attribute AIC.
- data The effectively analyzed data. If density of points is too high, e.g. with BreathId devices, data are subsampled before fitting.
- stan_fit The Stan fit for use with shinystan::launch_shiny or extraction of chains.

See Also

Base methods coef, plot, print; methods from package broom: tidy, augment.

Examples

```
library(breathtestcore)
suppressPackageStartupMessages(library(dplyr))
d = simulate_breathtest_data(n_records = 3) # default 3 records
data = cleanup_data(d$data)
# Use more than 80 iterations and 4 chains for serious fits
fit = stan_fit(data, chains = 1, iter = 80)
plot(fit) # calls plot.breathtestfit
# Extract coefficients and compare these with those
# used to generate the data
options(digits = 2)
cf = coef(fit)
cf %>%
  filter(grepl("m|k|beta", parameter )) %>%
  select(-method, -group) %>%
  tidyr::spread(parameter, value) %>%
  inner_join(d$record, by = "patient_id") %>%
  select(patient_id, m_in = m.y, m_out = m.x,
         beta_in = beta.y, beta_out = beta.x,
         k_in = k.y, k_out = k.x)
# For a detailed analysis of the fit, use the shinystan library

library(shinystan)
# launch_shinystan(fit$stan_fit)

# The following plots are somewhat degenerate because
# of the few iterations in stan_fit
suppressPackageStartupMessages(library(rstan))
stan_plot(fit$stan_fit, pars = c("beta[1]","beta[2]","beta[3]"))
stan_plot(fit$stan_fit, pars = c("k[1]","k[2]","k[3]"))
stan_plot(fit$stan_fit, pars = c("m[1]","m[2]","m[3]"))
```

`stan_group_fit` *Bayesian Stan fit to 13C Breath Data in Multiple Groups*

Description

Fits exponential beta curves to 13C breath test series data using Bayesian Stan methods, by assuming fixed between group effects. This model is overly parsiomonious. Do not use it unless you check the results carefully and understand why fits can be very bad.

Usage

```
stan_group_fit(data, dose = 100, sample_minutes = 15, student_t_df = 10,
  chains = 2, iter = 1000, model = "breath_test_group_1")
```

Arguments

data	Data frame or tibble as created by cleanup_data, with mandatory columns patient_id, group, minute and pdr. It is recommended to run all data through cleanup_data which will insert dummy columns for patient_id and minute if the data are distinct, and report an error if not. Since the Bayesian method is stabilized by priors, it is possible to fit single curves.
dose	Dose of acetate or octanoate. Currently, only one common dose for all records is supported.
sample_minutes	If mean sampling interval is < sampleMinutes, data are subsampled using a spline algorithm
student_t_df	When student_t_df < 10, the student distribution is used to model the residuals. Recommended values to model typical outliers are from 3 to 6. When student_t_df >= 10, the normal distribution is used.
chains	Number of chains for Stan
iter	Number of iterations for each Stan chain
model	Name of model; use names(stanmodels) for other models.

Value

A list of classes "breathteststangroupfit", "breathteststanfit" and "breathtestfit" with elements

- coef Estimated parameters as data frame in a key-value format with columns patient_id, group, parameter, metho
  and value. Has an attribute AIC.
- data The effectively analyzed data. If density of points is too high, e.g. with BreathId devices, data are subsampled before fitting.
- stan_fit The Stan fit for use with shinystan::launch_shiny or extraction of chains.

See Also

Base methods coef, plot, print; methods from package broom: tidy, augment.

Examples

```
library(breathtestcore)
library(dplyr)
data("usz_13c", package = "breathtestcore")
data = usz_13c %>%
  dplyr::filter( patient_id %in%
       c("norm_001", "norm_002", "norm_003", "norm_004",
         "pat_001", "pat_002","pat_003")) %>%
  cleanup_data()
fit = stan_group_fit(data, chains = 1, iter = 100)
plot(fit) # calls plot.breathtestfit
coef(fit)
```

Index