Package 'baclava'

July 22, 2025

Type Package

Title Bayesian Analysis of Cancer Latency with Auxiliary Variable Augmentation

Version 1.1

Date 2025-05-04

Description A novel data-augmentation Markov chain Monte Carlo sampling algorithm to fit a progressive compartmental model of disease in a Bayesian framework Morsomme, R.N., Holloway, S.T., Ryser, M.D. and Xu J. (2024) <doi:10.48550/arXiv.2408.14625>.

License GPL-3

Depends R (>= 4.1.0)

Imports Rcpp, RcppNumerical, ggplot2, coda, dplyr, stats, tibble, tidyr, foreach, doParallel

LinkingTo Rcpp, RcppEigen, RcppNumerical

Encoding UTF-8

NeedsCompilation yes

LazyData true

Suggests rmarkdown, knitr

Maintainer Shannon T. Holloway <shannon.t.holloway@gmail.com>

Author Raphael N. Morsomme [aut], Shannon T. Holloway [aut, cre], Jason Xu [ctb], Marc D. Ryser [ctb]

RoxygenNote 7.3.2

Collate 'RcppExports.R' 'aloocv.R' 'utilities.R' 'cohortODX.R' 'computeEndpoints.R' 'fit_baclava.R' 'predictODX_helpers.R' 'predictODX.R' 's3_methods.R' 'screen_data.R'

Repository CRAN

Date/Publication 2025-05-04 17:20:02 UTC

Contents

`all_cause_rates` *All Cause Mortality Rates*

Description

All-cause mortality rates for both male and female, male only, and female only for the 1974 birth cohort. Data were taken from the CDC Life Tables. Vital Statistics of the United States, 1974 Life Tables, Vol. II, Section 5. 1976.

Usage

```
data(all_cause_rates)
```

Format

`all_cause_rates` is a data.frame containing the following

- Age: An integer.
- both: A numeric. All-cause mortality rate for combined male and female.
- male: A numeric. Male only all-cause mortality rate.
- female: A numeric. Female only all-cause mortality rate.

`aloocv` *Approximate Leave-One-Out Cross-Validation*

Description

Approximate leave-one-out cross-validation computed from the posterior draws of the Markov chain Monte Carlo sampler as implemented in fit_baclava().

Usage

```
aloocv(
  object,
  data.clinical,
  data.assess,
  J.increment = 75L,
  J.max = 225L,
  ess.target = 50L,
  n.core = 1L,
  verbose = TRUE,
  lib = NULL
)
```

Arguments

`object`	The value object returned by fit_baclava().
`data.clinical`	A data.frame object. The clinical data on which the model is assessed. The data must be structured as for `fit_baclava()`; specifically, it must contain • id: A character, numeric, or integer object. The unique participant id to which the record pertains. Note these must include those provided in `data.assess`. Must be only 1 record for each participant. • age_entry: A numeric object. The age at time of entry into the study. Note that this data is used to calculate the normalization; to expedite numerical integration, it is recommended that the ages be rounded to minimize repeated calculations. Optional input 'round.age.entry' can be set to FALSE if this approximation is not desired; however, the computation time will significantly increase. • endpoint_type: A character object. Must be one of {"clinical", "censored", "preclinical"}. Type "clinical" indicates that disease was diagnosed in the clinical compartment (i.e., symptomatic). Type "preclinical" indicates that disease was diagnosed in the pre-clinical compartment (i.e., during an assessment). Type "censored" indicates participant was censored. • age_endpoint: A numeric object. The participant's age at the time the endpoint was evaluated. If the sensitivity parameter (beta) is arm-specific, an additional column `arm` is required indicating the study arm to which each participant is assigned. Similarly, if the preclinical Weibull distribution is group-specific, an additional column `grp.rateP` is required. See Details for further information.
`data.assess`	A data.frame object. The disease status assessment data on which the model is assessed. The data must be structured as for `fit_baclava()`; specifically, the data must contain • id: A character, numeric, or integer object. The unique participant id to which the record pertains. • age_assess: A numeric object. The participant's age at time of assessment. • disease_detected: An integer object. Must be binary 0/1, where 1 indicates that disease was detected at the assessment; 0 otherwise.

	If the sensitivity parameter (beta) is screen-type specific, an additional column `screen_type` is required indicating the type of each screen.
`J.increment`	An integer object. The number of replicates of each participant to generate in each iteration of the importance sampling procedure to attain desired effective sample size.
`J.max`	An integer object. The maximum number of samples to be drawn.
`ess.target`	An integer object. The target effective sample size in the importance sampling procedure.
`n.core`	An integer object. The function allows for the outer loop across participants to be run in parallel using foreach().
`verbose`	A logical object. If TRUE, progress information will be printed. This input will be ignored if n.core > 1.
`lib`	An optional character vector allowing for library path to be provided to cluster.

Details

Computes the predictive fit of a model. For each individual and each MCMC draw, the function approximates the marginal likelihood via importance sampling. It samples J.increment values of the individual's latent variables using the Metropolis-Hastings proposal distributions and computes the effective sample size (ESS) of the importance sampling procedure. If the target ESS is not met, J.increment additional samples are taken, and the ESS is re-evaluated. This is repeated until either the ESS is satisfied or J.max samples have been drawn.

Value

A list object. Element `summary` contains the min, mean, and the 1 likelihood; and the individual-level and estimated predictive fit. Element `result` contains the likelihood, ESS, and J for each MCMC sample for each participant.

Examples

```
data(screen_data)

theta_0 <- list("rate_H" = 7e-4, "shape_H" = 2.0,
                "rate_P" = 0.5  , "shape_P" = 1.0,
                "beta" = 0.9, psi = 0.4)
prior <- list("rate_H" = 0.01, "shape_H" = 1,
              "rate_P" = 0.01, "shape_P" = 1,
              "a_psi" = 1/2 , "b_psi" = 1/2,
              "a_beta" = 38.5, "b_beta" = 5.8)

# This is for illustration only -- the number of MCMC samples should be
# significantly larger and the epsilon values should be tuned.
example <- fit_baclava(data.assess = data.screen,
                       data.clinical = data.clinical,
                       t0 = 30.0,
                       theta_0 = theta_0,
                       prior = prior,
                       thin = 10L)
```

```
res <- aloocv(example, data.clinical, data.screen)
```

`cohortODX` *Cohort-specific Overdiagnosis*

Description

Estimates the overall and screening specific overdiagnosis probability for the cohort of the original analysis.

Usage

```
cohortODX(
  object,
  data.clinical,
  data.assess,
  other.cause.rates = NULL,
  plot = TRUE
)
```

Arguments

`object`	A 'baclava' object. The value object returned by fit_baclava().
`data.clinical`	A data.frame object. The clinical data. The data must be structured as • `id`: A character, numeric, or integer object. The unique participant id to which the record pertains. Note these must include those provided in `data.assess`. Must be only 1 record for each participant. • `age_entry`: A numeric object. The age at time of entry into the study. Note that this data is used to calculate a normalization; to expedite numerical integration, it is recommended that the ages be rounded. Optional input `round.age.entry` can be set to FALSE if this approximation is not desired; however, the computation time will significantly increase. • `endpoint_type`: A character object. Must be one of {"clinical", "censored", "preclinical"}. Type "clinical" indicates that disease was diagnosed in the clinical compartment (i.e., symptomatic). Type "preclinical" indicates that disease was diagnosed in the preclinical compartment (i.e., during an assessment). Type "censored" indicates disease was not diagnosed prior to end of study. • `age_endpoint`: A numeric object. The participant's age at the time the endpoint was evaluated. If the sensitivity parameter (beta) is arm-specific, an additional column `arm` is required indicating the study arm to which each participant is assigned. Similarly, if the preclinical Weibull distribution is group-specific, an additional column `grp.rateP` is required. See Details for further information. This input should be identical to that provided to obtain `object`.

`data.assess` A data.frame object. Disease status assessments recorded during healthy or preclinical compartment, e.g., screenings for disease. The data must be structured as

- `id`: A character, numeric, or integer object. The unique participant id to which the record pertains. Multiple records for each id are allowed.
- `age_assess`: A numeric object. The participant's age at time of assessment.
- `disease_detected`: An integer object. Must be binary 0/1, where 1 indicates that disease was detected at the assessment; 0 otherwise.

If the sensitivity parameter (beta) is screen-specific, an additional column `screen_type` is required indicating the type of each screen. This input should be identical to that provided to obtain `object`.

`other.cause.rates` A data.frame object. Age specific incidence rates that do not include the disease of interest. Must contain columns "Rate" and "Age".

`plot` A logical object. If TRUE, generates a boxplot of the overdiagnosis probability for each individual as a function of the screen at which disease was detected. Includes only the consecutive screens for which more than 1% of the screen detected cases were detected.

Value

A list object.

- `all` An n x S matrix containing the estimated overdiagnosis probability for each individual (n) and each posterior parameter set (S).
- `mean.individual` A vector containing the mean across S of the estimated overdiagnosis for each individual, i.e., `rowMeans(all)`.
- `mean.overall` A numeric, the mean overdiagnosis probability across all posterior parameter sets and screen-detected cases, i.e., `mean(all)`.
- `summary.by.screen` A matrix containing the summary statistics of `mean.individual` for the individuals detected positive at each screen, i.e., `summary(mean.individual[diagnosis_screen_id == i])`.

Examples

```
data(screen_data)

theta_0 <- list("rate_H" = 7e-4, "shape_H" = 2.0,
                "rate_P" = 0.5  , "shape_P" = 1.0,
                "beta" = 0.9, psi = 0.4)
prior <- list("rate_H" = 0.01, "shape_H" = 1,
              "rate_P" = 0.01, "shape_P" = 1,
              "a_psi" = 1/2 , "b_psi" = 1/2,
              "a_beta" = 38.5, "b_beta" = 5.8)

# This is for illustration only -- the number of Gibbs samples should be
# significantly larger and the epsilon values should be tuned.
```

```
example <- fit_baclava(data.assess = data.screen,
                       data.clinical = data.clinical,
                       t0 = 30.0,
                       theta_0 = theta_0,
                       prior = prior,
                       save.latent = TRUE)

# if rates are not available, an all cause dataset is provided in the package
# NOTE: these predictions will be over-estimated

data(all_cause_rates)
all_cause_rates <- all_cause_rates[, c("Age", "both")]
colnames(all_cause_rates) <- c("Age", "Rate")

cohort_odx <- cohortODX(object = example,
                        data.clinical = data.clinical,
                        data.assess = data.screen,
                        other.cause.rates = all_cause_rates,
                        plot = FALSE)
```

`fit_baclava` *Bayesian Analysis of Cancer Latency with Auxiliary Variable Augmentation*

Description

Markov chain Monte Carlo sampler to fit a three-state mixture compartmental model of cancer natural history to individual-level screening and cancer diagnosis histories in a Bayesian framework.

Usage

```
fit_baclava(
  data.assess,
  data.clinical,
  baclava.object = NULL,
  M = 100L,
  thin = 1L,
  t0 = 0,
  theta_0 = list(),
  prior = list(),
  epsilon_rate_H = 0.001,
  epsilon_rate_P = 0.001,
  epsilon_psi = 0.001,
  indolent = TRUE,
  adaptive = NULL,
  round.age.entry = TRUE,
  verbose = TRUE,
  save.latent = FALSE
```

```
)

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

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

Arguments

data.assess	A data.frame. Disease status assessments recorded during healthy or preclinical compartment, e.g., screenings for disease. The data must be structured as • id: A character, numeric, or integer object. The unique participant id to which the record pertains. Multiple records for each id are allowed. • age_assess: A numeric object. The participant's age at time of assessment. • disease_detected: An integer object. Must be binary 0/1, where 1 indicates that disease was detected at the assessment; 0 otherwise. If the sensitivity parameter (beta) is screen-specific, an additional column screen_type is required indicating the type of each screen.
data.clinical	A data.frame. The clinical data. The data must be structured as • id: A character, numeric, or integer object. The unique participant id to which the record pertains. Note these must include those provided in data.assess. Must be only 1 record for each participant. • age_entry: A numeric object. The age at time of entry into the study. Note that this data is used to calculate a normalization; to expedite numerical integration, it is recommended that the ages be rounded. Optional input round.age.entry can be set to FALSE if this approximation is not desired; however, the computation time will significantly increase. • endpoint_type: A character object. Must be one of {"clinical", "censored", "preclinical"}. Type "clinical" indicates that disease was diagnosed in the clinical compartment (i.e., symptomatic). Type "preclinical" indicates that disease was diagnosed in the preclinical compartment (i.e., during an assessment). Type "censored" indicates disease was not diagnosed prior to end of study. • age_endpoint: A numeric object. The participant's age at the time the endpoint was evaluated. If the sensitivity parameter (beta) is arm-specific, an additional column arm is required indicating the study arm to which each participant is assigned. Similarly, if the preclinical Weibull distribution is group-specific, an additional column grp.rateP is required. See Details for further information.
baclava.object	NULL or a 'baclava' object. To continue a calculation, provide the object returned by a previous call.
M	A positive integer object. The number of Monte Carlo samples. This is the total, i.e., M = adaptive$warmup + n_MCMC.

thin	A positive integer object. Keep each thin-th step of the sampler after the warmup period, if any, is complete.
t0	A non-negative scalar numeric object. The risk onset age. Must be less than the earliest assessment age, entry age, and endpoint age. If baclava.object is a 'baclava' object, this input is ignored.
theta_0	A list object. The initial values for all distribution parameters. If baclava.object is a 'baclava' object, this input is ignored. See Details for further information.
prior	A list object. The prior parameters. If baclava.object is a 'baclava' object, this input is ignored. See Details for further information.
epsilon_rate_H	A small scalar numeric. The Monte Carlo step size for rate_H (the rate parameter of the Weibull of the healthy compartment). If baclava.object is a 'baclava' object, this input is ignored.
epsilon_rate_P	A small scalar numeric or named numeric vector. The Monte Carlo step size for rate_P (the rate parameter of the Weibull of the preclinical compartment). If group-specific Weibull distributions are used, this must be a vector; see Details for further information. If baclava.object is a 'baclava' object, this input is ignored.
epsilon_psi	A small scalar numeric. The Monte Carlo step size for parameter psi (the probability of indolence). If disease under analysis does not have an indolent state, set to 0 and ensure that the initial value for psi in theta_0 is also 0. If baclava.object is a 'baclava' object, this input is ignored.
indolent	A logical object. If FALSE, disease under analysis does not have an indolent state, i.e., it is always progressive. This input is provided for convenience; if FALSE, epislon_psi and theta_0$psi will be set to 0. If baclava.object is a 'baclava' object, this input is ignored.
adaptive	NULL or named list. If NULL, the step sizes are not modified in the MCMC. If a list, the parameters for the adaptive MCMC. The provided list must contain elements "delta", the target acceptance rate; "warmup", the number of iterations to apply step size correction; and parameters "m0", "kappa", and "gamma". See Details for further information. If baclava.object is a 'baclava' object, this input is ignored.
round.age.entry	A logical object. If TRUE, the age at time of entry will be rounded to the nearest integer prior to performing the MCMC. This data is used to estimate the probability of experiencing clinical disease prior to entering the study, which is estimated using a time consuming numerical integration procedure. It is expected that rounding the ages at time of entry introduces minimal bias. If FALSE, and ages cannot be grouped, these integrals significantly increase computation time. If baclava.object is a 'baclava' object, this input is ignored.
verbose	A logical object. If TRUE, a progress bar will be shown during the MCMC.
save.latent	A logical object. If TRUE, latent variable tau_HP and indolence will be returned. These can be very large matrices. To estimate the cohort overdiagnosis probability using cohortODX(), this must be set to TRUE.
object	An object of class baclava.
...	Ignored.
x	An object of class baclava.

Details

Input `theta_0` contains the initial values for all distribution parameters. The list must include

- `rate_H`: A scalar numeric. The rate for the Weibull distribution of the healthy compartment.
- `shape_H`: A scalar numeric. The shape parameter for the Weibull distribution of the healthy compartment.
- `rate_P`: A numeric scalar or named numeric vector. The rate parameter for each Weibull distribution of the preclinical compartment. If all participants follow the same Weibull distribution, provide a scalar. If multiple preclinical Weibull distributions are used, see note below.
- `shape_P`: A scalar numeric. The shape parameter for all Weibull distributions of the preclinical compartment.
- `beta`: A scalar numeric or named numeric vector. The assessment sensitivity. If the sensitivity is the same for all participants, provide a scalar. If the sensitivity is arm- or screen-type-specific, see note below. Each element must be in [0, 1].
- `psi`: A scalar numeric. The probability of being indolent. Must be in [0,1]. If disease is always progressive, this element is required, but its value must be set to 0.

Input `prior` contains all distribution parameters for the priors. The list must include

- `rate_P`: A scalar numeric or named vector object. The rate for the Gamma(shape_P, rate_P) prior on the rate of the Weibull of the preclinical compartment. If group-specific distributions are used, see note below.
- `shape_P`: A scalar numeric or named vector object. The shape for the Gamma(shape_P, rate_P) prior on the rate of the Weibull of the preclinical compartment. If group-specific distributions are used, see note below.
- `rate_H`: A scalar numeric. The rate for the Gamma(shape_H, rate_H) prior on the rate of the Weibull of the healthy compartment.
- `shape_H`: A scalar numeric. The shape for the Gamma(shape_H, rate_H) prior on the rate of the Weibull of the healthy compartment.
- `a_beta`: A positive scalar numeric or named numeric vector. The first parameter of the Beta(a, b) prior on the assessment sensitivity. If arm- or screen-type-specific distributions are used, see note below. If beta is not allowed to change, specify 0.0.
- `b_beta`: A positive scalar numeric or named numeric vector. The second parameter of the Beta(a, b) prior on the assessment sensitivity. If arm- or screen-type-specific distributions are used, see note below. If beta is not allowed to change, specify 0.0.
- `a_psi`: A positive scalar numeric. The first parameter of the Beta(a, b) prior on the indolence probability. If disease under analysis does not have an indolent state, this element must be included, but it will be ignored.
- `b_psi`: A positive scalar numeric. The second parameter of the Beta(a, b) prior on the indolence probability. If disease under analysis does not have an indolent state, this element must be included, but it will be ignored.

It is possible to assign participants to study arms such that each arm has its own screening sensitivities and/or rate_P distributions, or to assign screen-type specific sensitivities.

To designate study arms, each of which will have its own screening sensitivities:

- Provide an additional column in `data.clinical` named "arm", which gives the study arm to which each participant is assigned. For example, `data.clinical$arm = c("Control", "Tx", "Tx", ...)`.
- Define all beta related prior parameters as named vectors. For example, `prior$a_beta = c("Control" = 1, "Tx" = 38.5)`, and `prior$b_beta = c("Control" = 1, "Tx" = 5.8)`
- Define the initial beta values of theta as a named vector. For example, `theta_0$beta = c("Control" = 0.75, "Tx" = 0.8)`.

Similarly, if using multiple preclinical Weibull distributions (distributions will have the same shape_P),

- Provide an additional column in `data.clinical` named "grp.rateP", which assigns each participant to one of the preclinical Weibull distributions. For example, `data.clinical$grp.rateP = c("rateP1", "rateP2", "rateP2", ... )`.
- Define the rate_P prior parameter as a named vector. For example, `prior$rate_P <- c("rateP1" = 0.01, "rateP2" = 0.02)`.
- Define the shape_P prior parameter as a named vector. For example, `prior$shape_P <- c("rateP1" = 1, "rateP2" = 2)`.
- Define the initial rate_P values of theta as a named vector. For example, `theta_0$rate_P <- c("rateP1" = 1e-5, "rateP2" = 0.01)`.
- Define step size of rate_P as a named vector. For example, `epsilon_rate_P <- c("rateP1" = 0.001, "rateP2" = 0.002)`.

To assign screen-specific sensitivities,

- Provide an additional column in `data.assess` named "screen_type", which gives the screening type for each screen. For example, `data.assess$screen_type = c("film", "2D", "2D", ...)`.
- Define all beta related prior parameters as named vectors. For example, `prior$a_beta = c("film" = 1, "2D" = 38.5)`, and `prior$b_beta = c("film" = 1, "2D" = 5.8)`
- Define the initial beta values of theta as a named vector. For example, `theta_0$beta = c("film" = 0.75, "2D" = 0.8)`.

NOTE: If using integers to indicate group membership, vector names still must be provided. For example, if group membership is binary 0/1, vector elements of the prior, initial theta, and step size must be named as "0" and "1".

The adaptive MCMC tuning expression at step m + 1 is defined as

$$\epsilon_{m+1} = (1 - m^{\kappa})\epsilon_m + m^{\kappa}\xi_{m+1},$$

where

$$\xi_{m+1} = \frac{\sqrt{m}}{\gamma}\frac{1}{m + m_0}\sum_{i=1}^{m}(\alpha_m - \delta).$$

To initiate the adaptive selection procedure, input `adaptive` must specify the parameters of the above expressions. Specifically, the provided list must contain elements "delta", the target acceptance rate; "warmup", the number of iterations to apply step size correction; and parameters "m0", "kappa", and "gamma".

Value

An object of S3 class `baclava`, which extends a list object.

- theta: A list of the posterior distribution parameters at the thinned samples.
  - rate_H: A numeric vector. The rates for the Weibull of the the healthy compartment.
  - shape_H: A scalar numeric. The input shape_H parameter.
  - rate_P: A numeric matrix. The rates for the Weibull of the preclinical compartment.
  - shape_P: A scalar numeric. The input shape_P parameter.
  - beta: A numeric matrix. The assessment sensitivities.
  - psi: A numeric vector. The probabilities of indolence. Will be NA if disease is always progressive.
- tau_hp: If `save.latent = TRUE`, a matrix. The age at time of transition from healthy to preclinical compartment for each participant at the thinned samples.
- indolent: If `save.latent = TRUE`, a matrix. The indolent status for each participant at the thinned samples. Will be `NA` if disease is always progressive.
- accept: A list of the accept indicator at the thinned samples.
  - rate_H: A numeric vector.
  - rate_P: A numeric matrix.
  - tau_hp: If `save.latent = TRUE`, a matrix. Will be NA if current and new transition ages are Inf.
  - psi: A numeric vector. The probability of indolence. Will be `NA` if disease is always progressive.
- epsilon: A list. The step sizes for each parameter.
- adaptive: A list. Settings for the adaptive procedure. Will be NA if adaptive procedure not requested.
- last_theta: A list. The theta parameters of the last MCMC iteration.
- prior: A list. The provided parameters of the prior distributions.
- setup: A list of inputs provided to the call.
  - t0: The input age of risk onset.
  - indolent: TRUE if disease is not progressive.
  - round.age.entry: TRUE if age at entry was rounded to the nearest whole number.
  - groups.beta: A vector of the beta grouping values.
  - groups.rateP: A vector of the rate_P grouping values.
  - thin: The number of samples dropped between kept MCMC iterations.
  - initial.theta: theta_0 as provided by user.
  - initial.prior: prior as provided by user.
- clinical.groupings: A data.frame of the original data's arm/rateP grouping.
- screen_types: A data.frame of the original data's screen type grouping.
- call: The matched call.

Functions

- `summary(baclava)`: Summary statistics of posterior distribution parameters
- `print(baclava)`: Print summary statistics of posterior distribution parameters

Examples

```
data(screen_data)

theta_0 <- list("rate_H" = 7e-4, "shape_H" = 2.0,
                "rate_P" = 0.5  , "shape_P" = 1.0,
                "beta" = 0.9, psi = 0.4)
prior <- list("rate_H" = 0.01, "shape_H" = 1,
              "rate_P" = 0.01, "shape_P" = 1,
              "a_psi" = 1/2 , "b_psi" = 1/2,
              "a_beta" = 38.5, "b_beta" = 5.8)

# This is for illustration only -- the number of Gibbs samples should be
# significantly larger and the epsilon values should be tuned.
example <- fit_baclava(data.assess = data.screen,
                       data.clinical = data.clinical,
                       t0 = 30.0,
                       theta_0 = theta_0,
                       prior = prior)

summary(example)
print(example)

# To continue this calculation
example_continued <- fit_baclava(data.assess = data.screen,
                                 data.clinical = data.clinical,
                                 baclava.object = example)
```

---

plot.baclava *Plot Posterior Distribution Parameters*

Description

Convenience function to facilitate exploration of posterior distributions through trace plots, autocorrelations, and densities, as well as plotting the estimated hazard for transitioning to the preclinical compartment.

Usage

```
## S3 method for class 'baclava'
plot(
  x,
  y,
  ...,
```

```
  type = c("density", "trace", "acf", "hazard"),
  burnin = 0L,
  max_age = 90L,
  trace_var = c("psi", "rate_H", "rate_P", "beta")
)
```

Arguments

x	An object of class baclava.
y	Ignored
...	Ignored
type	A character object. One of {"density", "trace", "acf", "hazard"}. The type of plot to generate
burnin	An integer object. Optional. The number of burn-in samples. Used only for type = "trace". One trace plot is generated for the burnin iterations; a second for the post-burnin iterations. Note, this refers to the number of kept (thinned) samples.
max_age	A numeric object. For type = "hazard", the maximum age at which to evaluate the hazard.
trace_var	A character object. The parameter for which trace plots are to be generated. Must be one of {"psi", "rate_H", "rate_P", "beta"}

Value

A gg object

Examples

```
data(screen_data)

theta_0 <- list("rate_H" = 7e-4, "shape_H" = 2.0,
                "rate_P" = 0.5  , "shape_P" = 1.0,
                "beta" = 0.9, psi = 0.4)
prior <- list("rate_H" = 0.01, "shape_H" = 1,
              "rate_P" = 0.01, "shape_P" = 1,
              "a_psi" = 1/2 , "b_psi" = 1/2,
              "a_beta" = 38.5, "b_beta" = 5.8)

# This is for illustration only -- the number of Gibbs samples should be
# significantly larger and the epsilon values should be tuned.
example <- fit_baclava(data.assess = data.screen,
                       data.clinical = data.clinical,
                       t0 = 30.0,
                       theta_0 = theta_0,
                       prior = prior)

plot(example)
plot(example, type = "trace", trace_var = "psi", burnin = 0L)
plot(example, type = "trace", trace_var = "rate_H", burnin = 0L)
```

```
plot(example, type = "trace", trace_var = "rate_P", burnin = 0L)
plot(example, type = "trace", trace_var = "beta", burnin = 0L)
plot(example, type = "acf")
plot(example, type = "hazard", max_age = 70)
```

---

predictODX *Estimate the Overall and Per Screen Overdiagnosis Rates*

Description

Using the posterior parameter distributions, calculates the infinite population estimates of the probability of overdiagnosis at each screening episode due to indolence and/or death by other causes.

Usage

```
predictODX(
  object,
  screening.schedule,
  other.cause.rates,
  groups.rateP = NULL,
  screen.type = NULL,
  burnin = 1000L,
  verbose = TRUE
)

## S3 method for class 'baclava.ODX.pred'
plot(x, y, ...)
```

Arguments

`object`	An object of S3 class 'baclava'. The value object returned by `fit_baclava()`.
`screening.schedule`	A numeric vector object. A vector of ages at which screenings occur.
`other.cause.rates`	A data.frame object. Must contain columns "Rate" and "Age".
`groups.rateP`	An integer scalar object. If model included groups with different sojourn parameters, the group for which overdiagnosis is to be estimated. Must be one of `object$setup$groups.rateP`
`screen.type`	An integer scalar object. If model included screen-type, specific sensitivity parameters, the screen-type for which overdiagnosis is to be estimated. Must be one of `object$setup$groups.beta`
`burnin`	An integer object. Optional. The number of burn-in samples. Used only for `type = "trace"`. One trace plot is generated for the burnin iterations; a second for the post-burnin iterations. Note, this refers to the kept (thinned) samples.
`verbose`	A logical object. If TRUE, progress bars will be displayed.

x	A an object of S3 class 'baclava.PDX.pred' as returned by `predictODX()`.
y	Ignored.
...	Ignored.

Details

Provided birth cohort life table is an all cause tables obtained from the CDC Life Tables. Vital Statistics of the United States, 1974 Life Tables, Vol. II, Section 5. 1976. Estimated "other cause" mortality will thus be overestimated when using these tables. It is recommended that user provide data that has been corrected to exclude death due to the disease under analysis.

Value

A list object. For each screen in `screening.schedule`, a matrix providing the mean total overdiagnosis and the mean overdiagnosis due to indolent/progressive tumors, as well as their 95 Similarly, element `overall` provides these estimates for the full screening schedule.

Functions

- `plot(baclava.ODX.pred)`: Generate column plot of predicted overdiagnosis for each screen.

Examples

```
data(screen_data)

theta_0 <- list("rate_H" = 7e-4, "shape_H" = 2.0,
                "rate_P" = 0.5  , "shape_P" = 1.0,
                "beta" = 0.9, psi = 0.4)
prior <- list("rate_H" = 0.01, "shape_H" = 1,
              "rate_P" = 0.01, "shape_P" = 1,
              "a_psi" = 1/2 , "b_psi" = 1/2,
              "a_beta" = 38.5, "b_beta" = 5.8)

# This is for illustration only -- the number of MCMC samples should be
# significantly larger and the epsilon values should be tuned.
example <- fit_baclava(data.assess = data.screen,
                       data.clinical = data.clinical,
                       t0 = 30.0,
                       theta_0 = theta_0,
                       prior = prior)

# if rates are not available, an all cause dataset is provided in the package
# NOTE: these predictions will be over-estimated

data(all_cause_rates)
all_cause_rates <- all_cause_rates[, c("Age", "both")]
colnames(all_cause_rates) <- c("Age", "Rate")

# using single screen for example speed
predicted_odx <- predictODX(object = example,
                            other.cause.rates = all_cause_rates,
```

```
                            screening.schedule = 40,
                            burnin = 10)

plot(predicted_odx)
```

`screen_data` *Toy Dataset*

Description

This toy dataset is provided to facilitate examples and provide an example of the required input format. Though the data were simulated under a scenario similar to a real-world breast cancer screening trial, they should not be interpreted as representing true trial data.

Usage

```
data(screen_data)
```

Format

Two datasets are provided.

`data.screen` is a data.frame containing the following screening information for 89 participants (287 assessments)

- id: A character. Participant ids.
- age_assess: A numeric. The participant age as time of assessment.
- disease_detected: An integer. 1 = disease detected at assessment; 0 otherwise

`data.clinical` is a data.frame containing the following information for 89 participants.

- id: A character. Participant ids.
- age_entry: A numeric. The participant age as time of study entry.
- endpoint_type: A character. One of {"clinical", "preclinical", "censored"}, indicating if participant was diagnosed with the disease in the clinical compartment, was diagnosed in the pre-clinical compartment, or was censored.
- age_endpoint: A numeric. The participant's age at time the endpoint was ascertained.

Index