

Package: msmbayes (via r-universe)

September 18, 2026

Title Bayesian Multi-State Models for Intermittently-Observed Data

Version 0.3

Description Bayesian multi-state models for intermittently-observed data. Markov and phase-type semi-Markov models, and misclassification hidden Markov models.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0),

Imports abind, cli, dplyr, expm, ggplot2, glue, hardhat, methods, posterior, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rlang, rstan (>= 2.18.1), rstantools (>= 2.4.0), stringr, tidybayes, tidyr, vctrs

Suggests cmdstanr, msm, bayesplot, testthat (>= 3.0.0), rmarkdown, knitr, magick, pdftools, usethis, maxLik, lubridate, colorspace

LazyData true

LazyDataCompression gzip

VignetteBuilder knitr

URL <https://github.com/chjackson/msmbayes>

BugReports <https://github.com/chjackson/msmbayes/issues>

Additional_repositories <https://mc-stan.org/r-packages>

Config/testthat/edition 3

Biarch true

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

Config/Needs/website rmarkdown

Config/pak/sysreqs cmake make libicu-dev
Repository <https://chjackson.r-universe.dev>
Date/Publication 2026-05-06 15:37:49 UTC
RemoteUrl <https://github.com/chjackson/msmbayes>
RemoteRef HEAD
RemoteSha 80e2bd7f52b30594e7617c0c6e5e7fad595b25a0

Contents

msmbayes-package	3
bigdat	3
canpars_to_rates	4
edf	4
ematrix	5
example_models	6
illdeath_data	6
infsim	7
loghr	8
loglik	8
logoddsnext	9
logtaf	9
mean_sojourn	10
msmbayes	11
msmbayes_prior_sample	15
msmbayes_priorpred_sample	18
msmhists	21
msmhists_bardata	23
msmprior	25
n3_moment_bounds	27
nphase	28
nphase_Q	29
phaseapprox_pars	30
pmatrix	30
pmatrixdf	31
pnext	32
qdf	33
qmatrix	34
qphaseapprox	35
rrnextdoc	36
shape_in_bounds	36
shape_ubound	37
shapyscale_to_rates	38
sim_2state_smm	39
soj_prob	39
soj_quantile	40
standardise_to	41

msmbayes-package 3

statetable 42
summary.msmbayes 43
totlos 44

Index 46

msmbayes-package	<i>The 'msmbayes' package for Bayesian multi-state modelling of intermittently-observed data</i>
------------------	--

Description

For an introduction to and overview of the msmbayes package, and full documentation, see <http://chjackson.github.io/msmbayes>.

For more resources on multi-state modelling, see the [msm package](#) and its documentation.

Author(s)

Maintainer: Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk> ([ORCID](#))

See Also

Useful links:

- <https://github.com/chjackson/msmbayes>
- Report bugs at <https://github.com/chjackson/msmbayes/issues>

bigdat	<i>A simulated multistate dataset with lots of observations and covariates</i>
--------	--

Description

A simulated multistate dataset with lots of observations and covariates

Usage

bigdat

Format

See data-raw/bigdata.R in the source for simulation settings.

canpars_to_rates	<i>Convert between canonical parameters and rates for a phase-type distribution</i>
------------------	---

Description

Convert between canonical parameters and rates for a phase-type distribution

Usage

```
canpars_to_rates(pars, type = "vector")
```

```
rates_to_canpars(rates, type = "vector")
```

Arguments

pars	Canonical parameters, supplied in the order: • sojourn rate in phase 1 • additive increments in sojourn rates for each successive phase • probabilities (not rates) of absorption from each phase, for phase 1 up to the second last. or a list with three components, one vector for each of these three parameter types.
type	"vector" or "list".
rates	List with two components for progression and absorption rates, in increasing order of phase, or a vector with these concatenated.

Value

A list with components

- p progression rates between phases
- a absorption rates

or a vector with these components concatenated, depending on the "type" argument.

edf	<i>Misclassification error probabilities from an msmbayes model</i>
-----	---

Description

Misclassification error probabilities from an msmbayes model

Usage

```
edf(draws)
```

Arguments

draws Object returned by [msmbayes](#).

Value

A data frame with one row per modelled misclassification probability. `from` indicates the true state, and `to` indicates the observed state. Error probabilities fixed by the user are not included.

See Also

[qdf](#) for more information about the format.

ematrix	<i>Matrix of misclassification error probabilities from an msmbayes model</i>
---------	---

Description

Matrix of misclassification error probabilities from an msmbayes model

Usage

```
ematrix(draws, type = "posterior")
```

Arguments

draws Object returned by [msmbayes](#).

type "posterior" to return rvar objects containing posterior samples.
 "mode" to return the output evaluated at the posterior mode of the basic parameters (only applicable if model was fitted with posterior mode optimisation).

Value

An array or matrix of rvar objects containing the misclassification error matrix for each new prediction data point

example_models

Example fitted model objects used for testing msmbayes

Description

Example fitted model objects used for testing msmbayes

Usage

infsim_model

infsim_modelc

infsim_modelp

infsim_modelpc

Format

An object of class msmbayes, obtained by fitting a Markov model to the dataset [infsim](#). See `data-raw/infsim.R` in the source for the model specification code

infsim_modelc includes covariates on the transition intensities.

infsim_modelp is a phase-type model.

infsim_modelpc is a phase-type model with covariates.

Source

Simulated

illdeath_data

A simulated dataset from an illness-death model

Description

A simulated dataset from an illness-death model

Usage

illdeath_data

Format

See `data-raw/illdeath.R` in the source for simulation settings.

infsim	<i>Simulated infection testing data</i>
--------	---

Description

Simulated infection testing data

Usage

```
infsim
```

```
infsim2
```

Format

`infsim` has 3600 rows, with 36 state observations for each of 100 people. A smaller dataset `infsim2` has only 360 rows, from 20 people, and is simulated using a sojourn time of 60 days in the test-negative state and 10 days in test-positive.

Columns are:

- `subject` Subject identifier
- `days` Observation time (in days)
- `months` Observation time (in months)
- `state` State simulated from a Markov model with no covariates
- `sex`: "male" or "female".
- `age10`: Age, in units of 10 years since age 50
- `statec`: State simulated from a Markov model with covariates
- `statep`: State simulated from a phase-type model (unused in any examples. See `data-raw/infsim.R` in the source for simulation settings)
- `statepc`: State simulated from a phase-type model with covariates (unused in any examples)

An object of class `data.frame` with 360 rows and 14 columns.

Details

The transition intensities used for the simulation are defined using a mean sojourn time of 180 days in the "test negative" state and 10 days in the "test positive" state.

For the model with covariates, the log hazard ratios are 2 for male on the 1-2 transition, 1 for `age10` on the 1-2 transition, and -1 for `age10` on the 2-1 transition. Baseline intensities are for female, age 50 (i.e. `age10=0`).

In the state data, state 1 is negative and 2 is positive.

Source

Simulated

loghr	<i>(Log) hazard ratios for covariates on transition intensities</i>
-------	---

Description

In semi-Markov models specified with `pastates`, these only include covariate effects on transitions out of "Markov" states, that is, those not included in `pastates`. Effects on scale parameters of sojourn distributions in "semi-Markov" states are extracted with `logtaf`.

Usage

```
loghr(draws)
```

```
hr(draws)
```

Arguments

`draws` Object returned by `msmbayes`.

Value

A data frame containing samples from the posterior distribution. See `qdf` for notes on this format and how to summarise. `hr` returns these on the natural scale, `loghr` returns them on the log scale.

loglik	<i>Log likelihood from an msmbayes model</i>
--------	--

Description

Log likelihood from an `msmbayes` model

Usage

```
loglik(draws)
```

```
## S3 method for class 'msmbayes'
logLik(object, ...)
```

Arguments

`draws` Object returned by `msmbayes`.
`object` Object returned by `msmbayes`.
`...` Further arguments passed to both `qdf`, `hr`, `loghr` and `edf`.

Value

For `loglik`, a data frame with rows for the log likelihood, log prior density and log posterior density, and columns for the posterior (as an `rvar` object) and the mode (only if optimisation was used to fit the model, rather than MCMC).

If `msmbayes` was called with `priors="mle"`, the maximised log posterior and log likelihood should be the same.

For `logLik` (note the different capitalisation), just the likelihood (mode if available, or posterior if not) is returned. This is a method for the generic `logLik` function in the `stats` package.

logoddsnext	<i>Summarise posteriors for log odds of transitions from phase-type states</i>
-------------	--

Description

Log odds of transition to a competing destination state, relative to baseline destination state. Only applicable to phase-type approximation models, specified with `pastates`.

Usage

```
logoddsnext(draws, new_data = NULL, keep_covid = FALSE)
```

Arguments

<code>draws</code>	Object returned by msmbayes .
<code>new_data</code>	Data frame with covariate values to predict for
<code>keep_covid</code>	(logical) Keep the integer column <code>covid</code> identifying unique covariate combinations.

See Also

`pnext`

logtaf	<i>(Log) time acceleration factors in semi-Markov models</i>
--------	--

Description

Extracts the covariate effects on scale parameters in phase-type approximation sojourn distributions in semi-Markov models. Note these are not hazard ratios for transitions on the observable state space. They are time acceleration factors, such that an increase in one covariate unit makes time pass at `taf` times the speed, such that the sojourn time is expected to reduce by $\exp(-taf)$. The log TAFs are labelled β in the paper and vignettes.

Usage

```
logtaf(draws)
```

```
taf(draws)
```

Arguments

draws Object returned by [msmbayes](#).

Details

See `loghr` for effects of covariates on transitions out of Markov states, which are log hazard ratios.

Value

A data frame containing samples from the posterior distribution. See [qdf](#) for notes on this format and how to summarise.

[taf](#) returns these on the natural scale, [logtaf](#) returns them on the log scale.

mean_sojourn	<i>Mean sojourn times from an msmbayes model</i>
--------------	--

Description

Mean sojourn times from an msmbayes model

Usage

```
mean_sojourn(draws, new_data = NULL, states = "obs", keep_covid = FALSE)
```

Arguments

draws	Object returned by msmbayes .
new_data	Data frame with covariate values to predict for
states	If states="obs" (or "observed") then this describes mean sojourn times in the observable states. For phase-type models this is not generally equal to the sum of the phase-specific mean sojourn times, because an individual may transition out of the state before progressing to the next phase. If states="phase" (or "true", or "latent") then for phase-type models, this describes mean sojourn times in the latent state space.
keep_covid	(logical) Keep the integer column covid identifying unique covariate combinations.

Value

A data frame containing samples from the posterior distribution. See [qdf](#) for notes on this format and how to summarise.

msmbayes

*Bayesian multi-state models for intermittently-observed data***Description**

Fit a multi-state model to longitudinal data consisting of intermittent observation of a discrete state. Bayesian, approximate Bayesian or maximum likelihood estimation is used, via the Stan software.

Usage

```
msmbayes(
  data,
  state = "state",
  time = "time",
  subject = "subject",
  Q,
  covariates = NULL,
  paststates = NULL,
  pafamily = "gamma",
  panphase = NULL,
  E = NULL,
  Efix = NULL,
  obstype = NULL,
  deathexact = FALSE,
  obstrue = NULL,
  censor_states = NULL,
  constraint = NULL,
  nphase = NULL,
  priors = NULL,
  probb_initstate = NULL,
  soj_priordata = NULL,
  fit_method = NULL,
  keep_data = FALSE,
  ...
)
```

Arguments

data	Data frame giving the observed data.
state	Character string naming the observed state variable in the data. This variable must either be an integer in 1,2,...,K, where K is the number of states, or a factor with these integers as level labels. If omitted, this is assumed to be "state".
time	Character string naming the observation time variable in the data. If omitted, this is assumed to be "time".
subject	Character string naming the individual ID variable in the data. If omitted, this is assumed to be "subject".

Q	Matrix indicating the transition structure. A zero entry indicates that instantaneous transitions from (row) to (column) are disallowed. An entry of 1 (or any other positive value) indicates that the instantaneous transition is allowed. The diagonal of Q is ignored. There is no need to "guess" initial values and put them here, as is sometimes done in <code>msm</code>. Initial values for fitting are determined by Stan from the prior distributions, and the specific values supplied for positive entries of Q are disregarded.
covariates	Specification of covariates on transition intensities. This should be a list of formulae, or a single formula. If a list is supplied, each formula should have a left-hand side that looks like $Q(r, s)$, and a right hand side defining the regression model for the log of the transition intensity from state r to state s. For example, <pre>covariates = list(Q(1,2) ~ age + sex, Q(2,1) ~ age)</pre> specifies that the log of the 1-2 transition intensity is an additive linear function of age and sex, and the log 2-1 transition intensity is a linear function of age. You do not have to list all of the intensities here if some of them are not influenced by covariates. If a single formula is supplied, this is assumed to apply to all intensities. If doing this, then take care with potential lack of identifiability of effects from sparse data. In models with phase-type approximated states (specified with <code>pastates</code>), covariates are modelled through an accelerated failure time model. The effect is a multiplier on the scale parameter of the sojourn distribution. The covariate then has an identical multiplicative effect on all rates of transition between phases for a given state. The left hand side of the formula should contain <code>scale</code> instead of <code>Q</code>. For example, if state 1 has a phase type approximation, but state 2 is Markov, then we might supply covariates as: <pre>covariates = list(scale(1) ~ age + sex, Q(2,1) ~ age)</pre> In models with phase-type approximations and competing exit states, covariates on the relative risk of different exit states are specified with a formula with <code>rrnext</code> on the left hand side. For example in a model where state 1 has a phase-type approximation, and the next state could be either 2 or 3, a linear model on the log relative risk of transition to 3 (relative to the baseline 2) might be specified as: <pre>covariates = list(scale(1) ~ age + sex, rrnext(1,3) ~ x + time)</pre> In phase-type models specified with <code>nphase</code>, or misclassification models (specified with <code>E</code>), covariates on transition intensities are specified with $Q()$, where the numbers inside $Q()$ refer to the latent state space.
pastates	This indicates which states (if any) are given a Weibull or Gamma sojourn distribution approximated by a phase-type model. Ignored if <code>nphase</code> is supplied.
pafamily	"weibull" or "gamma", indicating the approximated sojourn distribution in the phased state. Either a vector of the same length as <code>pastates</code>, or just one to apply to all states.

panphase	Number of phases to use for each state given a phase-type Gamma or Weibull approximation. Vector of same length as <code>pastates</code> . Using more phases allows a wider range of shape parameters, (see Figure 2 of the <code>msmbayes</code> paper) but does not create extra parameters. Defaults to 3.
E	By default, <code>msmbayes</code> fits a (non-hidden) Markov model. If <code>E</code> is supplied, then a Markov model with misclassification is fitted, a type of hidden Markov model. <code>E</code> should then be a matrix indicating the structure of allowed misclassifications, where rows are the true states, and columns are the observed states. A zero entry in row r and column s indicates that true state r cannot be observed as state s . A non-zero (r, s) entry indicates that true state r may be misclassified as s . The diagonal of <code>E</code> is ignored.
Efix	Misclassification probabilities in Markov models are commonly not identifiable from data, particularly if the data are intermittently observed. Instead of estimating them, a Markov model with misclassification can be specified by supplying assumed misclassification probabilities in the <code>Efix</code> argument. This is a matrix with same dimensions as <code>E</code> . Any non-zero entries of <code>Efix</code> indicate the fixed known value for the corresponding misclassification probability. The (r, s) entry of <code>Efix</code> is 0 for any error probabilities that are estimated from the data or not permitted.
obstype	Character string, giving a variable in the data which defines what a "row of the data" means. The variable must contain only the following values, which may be different in different rows: 1: Intermittent observation. The state is unknown between the previous observation and the current observation (other than any knowledge implied by the structure <code>Q</code> of permitted transitions). 2: Exact transition times. The state is constant at the previous observed value between the previous and current times in the data, and the transition to the current state is made exactly at the current time. 3: "Exact death times". A transition to the current state is made exactly at the current time, but the state in the period between the previous observation and this transition is unknown. Typical (but not necessary) for observations of death in epidemiological/clinical studies. This is the same feature as in the <code>msm</code> package. If omitted, then all observations are assumed to be intermittent, with <code>obstype</code> 1.
deathexact	Set to TRUE if death times are observed with the <code>obstype</code> 3 scheme. This is a shortcut for including an <code>obstype</code> variable with 3 in the positions with the absorbing state, and 1 elsewhere. If there are multiple absorbing states, then this is taken to only apply to the last of them in the state space - use an <code>obstype</code> variable if you want it to apply to all absorbing states.
obstrue	Only applicable to models with misclassification. A character string indicating a variable in the data whose value is 1 if the true state is known to equal the value in "state", and 0 otherwise.
censor_states	A named list indicating the codes used for "censored" states. This is used when there are observations that are known to be one of a subset of states, but it is not known which. The names of the list indicate codes that may appear in the "state" variable. The values of the corresponding component indicate the subset which is represented by the code. For example

	<code>censor_states = list("99" = c(2,3), "999" = c(3,4))</code> means that a code of 99 in the "state" variable indicates "state is either 2 or 3 at this time", and a code of 999 indicates "state is either 3 or 4". Note the names of the list must be quoted strings that are interpretable as integers, since the "state" variable must be an integer. In misclassification models, the subset refers to values of the true state if <code>obstrue</code> is 1, or the observed state if <code>obstrue</code> is 0. Unlike in <code>msm</code>, there is no <code>censor</code> argument, and a <code>censor_states</code> must be supplied if there are censored states.
constraint	Constraints that a covariate has an equal effect on a particular set of transition intensities. A list with one component for each covariate that has constraints. Each component is a list of sets (or a single set) of intensities where the effect of that covariate is equal. For example, to constrain the effect of age to be equal for transitions 1-2 and 2-3, and also equal for transitions 1-4 and 2-4, and the effect of sexMALE to be equal for transitions 1-2 and 2-3, specify <pre>constraint = list(age = list(c("1-2", "2-3"), c("1-4", "2-4")), sex = list(c("1-2", "2-3")))</pre> This is the same feature as in <code>msm</code>, but with an easier interface. In <code>msmbayes</code> it is only supported for standard Markov models, not semi-Markov, phase-type or misclassification models.
nphase	Only required for models with phase-type sojourn distributions specified directly (not through <code>pastates</code>). <code>nphase</code> is a vector with one element per state, giving the number of phases per state. This element is 1 for states that do not have phase-type sojourn distributions.
priors	A list specifying priors. Each component should be the result of a call to <code>msmprior</code>. Any parameters with priors not specified here are given default priors: normal with mean -2 and SD 2 for log intensities, normal with mean 0 and SD 10 for log hazard ratios, normal(0,1) for log odds parameters in misclassification models. In phase-type approximation models, the default priors are normal with mean 2, SD 2 for scale parameters (i.e. the log inverse of the default prior for the rate), normal(0, SD=0.5) truncated on the supported region for log shape parameters, and normal(0,1) for log odds of transition (relative to first exit state) in structures with competing exit states. See <code>msmprior</code> for more details. If only one parameter is given a non-default prior, a single <code>msmprior</code> call can be supplied here instead of a list. Maximum likelihood estimation can be performed by setting <code>priors="mle"</code>, and using <code>fit_method="optimize"</code>. This is equivalent to estimating the posterior mode with improper uniform priors on the unconstrained parameter space (i.e. positive parameters on the log scale). Uncertainty is then quantified by sampling from the multivariate normal defined by the Hessian at the mode. The sample can be summarised to produce confidence intervals, as in the <code>ci="normal"</code> method in the <code>msm</code> package. These are equivalent to credible intervals from a Laplace approximation to the posterior.
prob_initstate	Probabilities of true states at a person's first observation time in a misclassification or model. If supplied, this should be a matrix with a row for each individual

	subject, and a column for each true state, or a vector with one element for each state that is assumed to apply to all individuals. If not supplied, every person is assumed to be in state 1 with probability 1 in misclassification models, or phase 1 of the observed state with probability 1 in phase-type models. Note no warning is currently given if the first observed state would be impossible if the person was really in state 1. This applies to both misclassification models, and phase-type models where a person's first observed state is phased. If the first observed state is not phased or misclassified, then this is ignored.
soj_priordata	Synthetic data that represents prior information about the mean sojourn times. Experimental, undocumented feature.
fit_method	Quoted string specifying the algorithm to fit the model. "sample" uses NUTS/HMC MCMC, via <code>rstan::sampling()</code>. This is the default unless <code>priors="mle"</code>. Alternatives are "optimize" to use posterior mode optimization (with respect to parameters on the log scale) followed by Laplace approximation around the mode (via <code>rstan::optimizing()</code>). This is the default if <code>priors="mle"</code>. "variational" to use variational Bayes (via <code>rstan::vb()</code>). "pathfinder", to use the Pathfinder variational algorithm via <code>cmdstanr</code>. This requires <code>cmdstan</code> and <code>cmdstanr</code> to be installed. The first time this is run for a particular <code>msmbayes</code> model class, the Stan program for that class is compiled, which will take a extra minute or two. The next time, it will not need to be recompiled. This also assumes you have write permission to the place where <code>msmbayes</code> is installed.
keep_data	Store a copy of the cleaned data in the returned object. FALSE by default.
...	Other arguments to be passed to the function from <code>rstan</code> or <code>cmdstanr</code> that fits the model. Note that initial values are determined by sampling from the prior (after dividing the prior SD 5), not using Stan's default, but this can be overridden here (currently not documented - this needs knowledge of the Stan variable names and formats)

Value

A data frame in the draws format of the **posterior** package, containing draws from the posterior of the model parameters.

Attributes are added to give information about the model structure, and a class "msmbayes" is prepended.

See, e.g. `summary.msmbayes`, `qdf`, `hr`, and similar functions, to extract parameter estimates from the fitted model.

`msmbayes_prior_sample` *Generate a sample from the prior distribution in a msmbayes model*

Description

Called in the same way as [msmbayes](#). The data should still be supplied in this function, to ensure we are simulating from a valid [msmbayes](#) model, but it is sufficient to supply an empty data frame with no rows, and columns named as if we were fitting a model with the given priors.

Usage

```
msmbayes_prior_sample(
  data,
  state = "state",
  time = "time",
  subject = "subject",
  Q,
  covariates = NULL,
  paststates = NULL,
  pafamily = "weibull",
  panphase = NULL,
  nphase = NULL,
  E = NULL,
  priors = NULL,
  nsim = 1
)
```

Arguments

<code>data</code>	Data frame giving the observed data.
<code>state</code>	Character string naming the observed state variable in the data. This variable must either be an integer in 1,2,...,K, where K is the number of states, or a factor with these integers as level labels. If omitted, this is assumed to be "state".
<code>time</code>	Character string naming the observation time variable in the data. If omitted, this is assumed to be "time".
<code>subject</code>	Character string naming the individual ID variable in the data. If omitted, this is assumed to be "subject".
<code>Q</code>	Matrix indicating the transition structure. A zero entry indicates that instantaneous transitions from (row) to (column) are disallowed. An entry of 1 (or any other positive value) indicates that the instantaneous transition is allowed. The diagonal of Q is ignored. There is no need to "guess" initial values and put them here, as is sometimes done in <code>msm</code>. Initial values for fitting are determined by Stan from the prior distributions, and the specific values supplied for positive entries of Q are disregarded.
<code>covariates</code>	Specification of covariates on transition intensities. This should be a list of formulae, or a single formula. If a list is supplied, each formula should have a left-hand side that looks like $Q(r, s)$, and a right hand side defining the regression model for the log of the transition intensity from state r to state s.

For example,

```
covariates = list(Q(1,2) ~ age + sex, Q(2,1) ~ age)
```

specifies that the log of the 1-2 transition intensity is an additive linear function of age and sex, and the log 2-1 transition intensity is a linear function of age. You do not have to list all of the intensities here if some of them are not influenced by covariates.

If a single formula is supplied, this is assumed to apply to all intensities. If doing this, then take care with potential lack of identifiability of effects from sparse data.

In models with phase-type approximated states (specified with `pastates`), covariates are modelled through an accelerated failure time model. The effect is a multiplier on the scale parameter of the sojourn distribution. The covariate then has an identical multiplicative effect on all rates of transition between phases for a given state. The left hand side of the formula should contain `scale` instead of `Q`. For example, if state 1 has a phase type approximation, but state 2 is Markov, then we might supply covariates as:

```
covariates = list(scale(1) ~ age + sex, Q(2,1) ~ age)
```

In models with phase-type approximations and competing exit states, covariates on the relative risk of different exit states are specified with a formula with `rrnext` on the left hand side. For example in a model where state 1 has a phase-type approximation, and the next state could be either 2 or 3, a linear model on the log relative risk of transition to 3 (relative to the baseline 2) might be specified as:

```
covariates = list(scale(1) ~ age + sex, rrnext(1,3) ~ x + time)
```

In phase-type models specified with `nphase`, or misclassification models (specified with `E`), covariates on transition intensities are specified with `Q()`, where the numbers inside `Q()` refer to the latent state space.

<code>pastates</code>	This indicates which states (if any) are given a Weibull or Gamma sojourn distribution approximated by a phase-type model. Ignored if <code>nphase</code> is supplied.
<code>pafamily</code>	"weibull" or "gamma", indicating the approximated sojourn distribution in the phased state. Either a vector of the same length as <code>pastates</code> , or just one to apply to all states.
<code>panphase</code>	Number of phases to use for each state given a phase-type Gamma or Weibull approximation. Vector of same length as <code>pastates</code> . Using more phases allows a wider range of shape parameters, (see Figure 2 of the <i>msmbayes</i> paper) but does not create extra parameters. Defaults to 3.
<code>nphase</code>	Only required for models with phase-type sojourn distributions specified directly (not through <code>pastates</code>). <code>nphase</code> is a vector with one element per state, giving the number of phases per state. This element is 1 for states that do not have phase-type sojourn distributions.
<code>E</code>	By default, <i>msmbayes</i> fits a (non-hidden) Markov model. If <code>E</code> is supplied, then a Markov model with misclassification is fitted, a type of hidden Markov model. <code>E</code> should then be a matrix indicating the structure of allowed misclassifications, where rows are the true states, and columns are the observed states. A zero entry in row r and column s indicates that true state r cannot be observed as state s . A non-zero (r, s) entry indicates that true state r may be misclassified as s . The diagonal of <code>E</code> is ignored.

priors	A list specifying priors. Each component should be the result of a call to <code>msmprior</code>. Any parameters with priors not specified here are given default priors: normal with mean -2 and SD 2 for log intensities, normal with mean 0 and SD 10 for log hazard ratios, normal(0,1) for log odds parameters in misclassification models. In phase-type approximation models, the default priors are normal with mean 2, SD 2 for scale parameters (i.e. the log inverse of the default prior for the rate), normal(0, SD=0.5) truncated on the supported region for log shape parameters, and normal(0,1) for log odds of transition (relative to first exit state) in structures with competing exit states. See <code>msmprior</code> for more details. If only one parameter is given a non-default prior, a single <code>msmprior</code> call can be supplied here instead of a list. Maximum likelihood estimation can be performed by setting <code>priors="mle"</code>, and using <code>fit_method="optimize"</code>. This is equivalent to estimating the posterior mode with improper uniform priors on the unconstrained parameter space (i.e. positive parameters on the log scale). Uncertainty is then quantified by sampling from the multivariate normal defined by the Hessian at the mode. The sample can be summarised to produce confidence intervals, as in the <code>ci="normal"</code> method in the <code>msm</code> package. These are equivalent to credible intervals from a Laplace approximation to the posterior.
nsim	Number of samples to generate

Value

A data frame with one column per model parameter (on a transformed scale, e.g. log intensities), and one row per sample. The names are in the natural format as specified in `priors`.

An attribute `"stan_names"` contains the names of the corresponding parameters in the draws object that would be returned by `msmbayes` if this model were to be fitted to data. These are the names used internally by Stan, and not meant to be interpretable by users.

An attribute `"expand"` contains the same sample but with parameters for covariate effects referring to state transitions on the latent space. Used internally for posterior predictive sampling.

`msmbayes_priorpred_sample`

Generate a dataset from the prior predictive distribution in a msmbayes model

Description

This generates a single sample of parameters from the prior, then generates observed states from a multi-state model with those parameters. The `data` argument should contain the time and subject indicators at which states are to be simulated (by default), or the maximum observation time (if `complete_obs=FALSE`).

Usage

```
msmbayes_priorpred_sample(
  data,
  state = "state",
  time = "time",
  subject = "subject",
  Q,
  covariates = NULL,
  paststates = NULL,
  pafamily = "gamma",
  panphase = NULL,
  nphase = NULL,
  E = NULL,
  priors = NULL,
  complete_obs = FALSE,
  cov_format = "orig"
)
```

Arguments

<code>data</code>	Data frame giving the observed data.
<code>state</code>	Character string naming the observed state variable in the data. This variable must either be an integer in 1,2,...,K, where K is the number of states, or a factor with these integers as level labels. If omitted, this is assumed to be "state".
<code>time</code>	Character string naming the observation time variable in the data. If omitted, this is assumed to be "time".
<code>subject</code>	Character string naming the individual ID variable in the data. If omitted, this is assumed to be "subject".
<code>Q</code>	Matrix indicating the transition structure. A zero entry indicates that instantaneous transitions from (row) to (column) are disallowed. An entry of 1 (or any other positive value) indicates that the instantaneous transition is allowed. The diagonal of Q is ignored. There is no need to "guess" initial values and put them here, as is sometimes done in <code>msm</code>. Initial values for fitting are determined by Stan from the prior distributions, and the specific values supplied for positive entries of Q are disregarded.
<code>covariates</code>	Specification of covariates on transition intensities. This should be a list of formulae, or a single formula. If a list is supplied, each formula should have a left-hand side that looks like $Q(r, s)$, and a right hand side defining the regression model for the log of the transition intensity from state r to state s. For example, <pre>covariates = list(Q(1,2) ~ age + sex, Q(2,1) ~ age)</pre> specifies that the log of the 1-2 transition intensity is an additive linear function of age and sex, and the log 2-1 transition intensity is a linear function of age. You

do not have to list all of the intensities here if some of them are not influenced by covariates.

If a single formula is supplied, this is assumed to apply to all intensities. If doing this, then take care with potential lack of identifiability of effects from sparse data.

In models with phase-type approximated states (specified with `pastates`), covariates are modelled through an accelerated failure time model. The effect is a multiplier on the scale parameter of the sojourn distribution. The covariate then has an identical multiplicative effect on all rates of transition between phases for a given state. The left hand side of the formula should contain scale instead of `Q`. For example, if state 1 has a phase type approximation, but state 2 is Markov, then we might supply covariates as:

```
covariates = list(scale(1) ~ age + sex, Q(2,1) ~ age)
```

In models with phase-type approximations and competing exit states, covariates on the relative risk of different exit states are specified with a formula with `rrnext` on the left hand side. For example in a model where state 1 has a phase-type approximation, and the next state could be either 2 or 3, a linear model on the log relative risk of transition to 3 (relative to the baseline 2) might be specified as:

```
covariates = list(scale(1) ~ age + sex, rrnext(1,3) ~ x + time)
```

In phase-type models specified with `nphase`, or misclassification models (specified with `E`), covariates on transition intensities are specified with `Q()`, where the numbers inside `Q()` refer to the latent state space.

<code>pastates</code>	This indicates which states (if any) are given a Weibull or Gamma sojourn distribution approximated by a phase-type model. Ignored if <code>nphase</code> is supplied.
<code>pafamily</code>	"weibull" or "gamma", indicating the approximated sojourn distribution in the phased state. Either a vector of the same length as <code>pastates</code> , or just one to apply to all states.
<code>panphase</code>	Number of phases to use for each state given a phase-type Gamma or Weibull approximation. Vector of same length as <code>pastates</code> . Using more phases allows a wider range of shape parameters, (see Figure 2 of the <code>msmbayes</code> paper) but does not create extra parameters. Defaults to 3.
<code>nphase</code>	Only required for models with phase-type sojourn distributions specified directly (not through <code>pastates</code>). <code>nphase</code> is a vector with one element per state, giving the number of phases per state. This element is 1 for states that do not have phase-type sojourn distributions.
<code>E</code>	By default, <code>msmbayes</code> fits a (non-hidden) Markov model. If <code>E</code> is supplied, then a Markov model with misclassification is fitted, a type of hidden Markov model. <code>E</code> should then be a matrix indicating the structure of allowed misclassifications, where rows are the true states, and columns are the observed states. A zero entry in row r and column s indicates that true state r cannot be observed as state s . A non-zero (r, s) entry indicates that true state r may be misclassified as s . The diagonal of <code>E</code> is ignored.
<code>priors</code>	A list specifying priors. Each component should be the result of a call to <code>msmprior</code> . Any parameters with priors not specified here are given default priors: normal with mean -2 and SD 2 for log intensities, normal with mean 0 and

SD 10 for log hazard ratios, normal(0,1) for log odds parameters in misclassification models.

In phase-type approximation models, the default priors are normal with mean 2, SD 2 for scale parameters (i.e. the log inverse of the default prior for the rate), normal(0, SD=0.5) truncated on the supported region for log shape parameters, and normal(0,1) for log odds of transition (relative to first exit state) in structures with competing exit states.

See [msmprior](#) for more details.

If only one parameter is given a non-default prior, a single `msmprior` call can be supplied here instead of a list.

Maximum likelihood estimation can be performed by setting `priors="mle"`, and using `fit_method="optimize"`. This is equivalent to estimating the posterior mode with improper uniform priors on the unconstrained parameter space (i.e. positive parameters on the log scale). Uncertainty is then quantified by sampling from the multivariate normal defined by the Hessian at the mode. The sample can be summarised to produce confidence intervals, as in the `ci="normal"` method in the `msm` package. These are equivalent to credible intervals from a Laplace approximation to the posterior.

<code>complete_obs</code>	If <code>complete_obs=FALSE</code> (the default) intermittently-observed states are generated for the subjects and times supplied in the <code>data</code> argument, using <code>msm::simmulti.msm</code>. The returned object is a data frame made by appending these states to <code>data</code>. If <code>complete_obs=TRUE</code>, one complete state transition history is generated using <code>msm::sim.msm</code>. The <code>data</code> argument should then consist of one row, with <code>time</code> giving the maximum observation time, and any covariates supplied, assumed to be time-constant. The returned object is a list.
<code>cov_format</code>	If <code>"orig"</code> the covariates are in their original form that they were supplied as. If <code>"design"</code> (or any other value) the covariates are returned as a design matrix, i.e. with factors converted to numeric contrasts.

Details

For phase-type approximation models, this simulates from the phase-type approximation, not the Weibull or Gamma (e.g) distribution that it is designed to approximate.

Value

A data frame or a list, see `msm::simmulti.msm` or `msm::sim.msm` respectively.

<code>msmhist</code>	<i>Illustrate the empirical distribution of states against time in intermittently-observed multistate data</i>
----------------------	--

Description

This works similarly to a histogram. The state observations are binned into time intervals with roughly equal numbers of observations. Within each bin, the probability $p(s)$ that an observation comes from each state s is estimated.

Usage

```
msmhist(
  data,
  state = "state",
  time = "time",
  subject = "subject",
  nbins,
  absorbing = NULL,
  censtimes = NULL,
  stacked = TRUE
)
```

Arguments

<code>data</code>	Data frame giving the observed data.
<code>state</code>	Character string naming the observed state variable in the data. This variable must either be an integer in 1,2,...,K, where K is the number of states, or a factor with these integers as level labels. If omitted, this is assumed to be "state".
<code>time</code>	Character string naming the observation time variable in the data. If omitted, this is assumed to be "time".
<code>subject</code>	Character string naming the individual ID variable in the data. If omitted, this is assumed to be "subject".
<code>nbins</code>	Number of time intervals to bin the state observations into. The underlying distribution of states illustrated by the plot will be assumed constant within each interval.
<code>absorbing</code>	Indices of any absorbing states. Individuals are assumed to stay in their absorbing state, and contribute one observation to each bin after their absorption time. By default, no states are assumed to be absorbing.
<code>censtimes</code>	Vector of maximum intended follow-up times for the people in the data who entered absorbing states. This supposes that had the person not entered the absorbing state, they would not have been observed after this time.
<code>stacked</code>	If TRUE do a bar chart with the probabilities for different states stacked on top of each other, so the y-axis spans 0 to 1 exactly. This is more compact. If FALSE, plot one panel per state, as is done in <code>prevalence.msm</code> . This is more convenient for constructing a check of the model fit.

Details

If each subject has at most one observation in a bin, then $p(s)$ is estimated as the proportion of observations in the bin that are of that state.

More generally, if an individual has more than one observation in the bin, $p(s)$ is estimated as follows. For each observed individual i and each state s , we define a variable $p(i, s)$ equal to the proportion of individual i 's observations that are of state s . For example, in a three-state model, where a person has two observations in a bin, and these are states 2 and 3, then $p(i, s) = 0, 0.5, 0.5$ for states 1, 2 and 3 respectively. The bin-specific estimate of $p(s)$ is then the average of $p(i, s)$ over individuals s who have at least one observation in that bin.

The results are visualised as a stacked bar plot. The individual observations of states are represented as points placed at random y positions within each state-specific bar.

This is intended as an alternative to the "observed prevalences" plot in the function `prevalence.msm` from the `msm` package, with a clearer connection to the data. It can be overlaid with predictions of transition probabilities from a `msmbayes` or `msm` model, to check the fit of the model.

The method used by "observed prevalences" plots places a strong assumption on the (unobserved) individual data, that individuals stay in the same state between observations, or transition at the midpoint between observations.

`msmhist` places no assumption on the individual data. Instead the assumption is placed on the distribution underlying the data. In a similar fashion to a histogram, it assumes that the distribution of states is the same at all times within each time interval bin.

Value

A `ggplot2` plot object.

See Also

[msmhist_bardata](#) to extract the numbers behind this plot so the plot can be customised by hand.

Examples

```
msmhist(infsim, "state", "months", "subject", nbins=30)
msmhist(infsim2, "state", "months", "subject", nbins=6)
```

<code>msmhist_bardata</code>	<i>Estimate state occupation probabilities to be illustrated by a bar plot in <code>msmhist</code></i>
------------------------------	--

Description

Estimate state occupation probabilities to be illustrated by a bar plot in `msmhist`

Usage

```
msmhist_bardata(
  data,
  state = "state",
  time = "time",
  subject = "subject",
  nbins,
  absorbing = NULL,
  censtimes = NULL
)
```

Arguments

data	Data frame giving the observed data.
state	Character string naming the observed state variable in the data. This variable must either be an integer in 1,2,...,K, where K is the number of states, or a factor with these integers as level labels. If omitted, this is assumed to be "state".
time	Character string naming the observation time variable in the data. If omitted, this is assumed to be "time".
subject	Character string naming the individual ID variable in the data. If omitted, this is assumed to be "subject".
nbins	Number of time intervals to bin the state observations into. The underlying distribution of states illustrated by the plot will be assumed constant within each interval.
absorbing	Indices of any absorbing states. Individuals are assumed to stay in their absorbing state, and contribute one observation to each bin after their absorption time. By default, no states are assumed to be absorbing.
censtimes	Vector of maximum intended follow-up times for the people in the data who entered absorbing states. This supposes that had the person not entered the absorbing state, they would not have been observed after this time.

Value

Data frame with one row per bin and state, and columns:

- binid: Integer ID for bin
- binlabel: Character label for bin, with time interval
- state: State
- binstart, binend: Start and end time of the bin (numeric)
- props: estimates of state s occupancy proportions $p(s)$ for each bin
- cumpstart, cumpend: Cumulative sum of props over the set of states, where cumpstart starts at 0, and cumpend ends at
 1. Intended for creating stacked bar plots with `geom_rect` or similar.

See Also

[msmhist](#)

msmprior

*Constructor for a prior distribution in msmbayes***Description**

Constructor for a prior distribution in msmbayes

Usage

```
msmprior(
  par,
  mean = NULL,
  sd = NULL,
  median = NULL,
  lower = NULL,
  upper = NULL
)
```

Arguments

par Character string indicating the model parameter to place the prior on. This should start with one of the following:

- "logq". Log transition intensity. It should then two include indices indicating the transition, e.g. "logq(2,3)" for the log transition intensity from state 2 to state 3.
- "q", Transition intensity (in the same format)
- "time". Defined as 1/q. This can be interpreted as the mean time to the next transition to state s for people in state r (from the point of view of someone observing one person at a time, and switching to observing a different person if a competing transition happens). The same format as logq and q with two indices.
- "loghr". Covariate effect on intensities of transition from states given a Markov model. The covariate name is supplied alongside the transition indices, e.g. "loghr(age,2,3)" for the effect of age on the log hazard ratio of transitioning from state 2 to state 3. For factor covariates, this should include the level, e.g. "loghr(sexMALE,2,3)" for level "MALE" of factor "sex".
- "logtaf". Covariate effect on the sojourn time in states given a semi-Markov model with a phase-type approximation. This is specified with only one index, indicating the state, e.g. "loghr(age,2)". Note this is interpreted as a log hazard ratio for times to transitions on the latent space, but for the sojourn time on the observable space, this is a "time acceleration factor", such that an coefficient of log(2) increases the risk of the next transition through halving the expected sojourn time.
- "hr". Hazard ratio.
- "loe" Log odds of error (relative to no misclassification). "loe(1,2)" indicates the log odds of misclassification in state 2 for true state 1, relative to no misclassification.

"logshape" "logscale" Log shape or scale parameter for the sojourn distribution in a phase-type approximation model. The index indicates the state, e.g. `logshape(2)` and `logscale(2)` indicate the log shape and scale parameter for the sojourn distribution in state 2.

"logoddsnext". Log odds of transition to a destination state in a phase-type approximation model with competing destination states. These parameters are only used in phase-type approximation models where there are multiple potential states that an individual could transition to immediately on leaving the state that has a phase-type approximation sojourn distribution. These parameters are defined with two indices. For example, `logoddsnext(1,2)` is the log odds of transition to state 2 on leaving state 1. The odds is the probability of transition to state 2 divided by the probability of transition to the first out of the set of potential destination states.

Covariate effects on competing transitions out of semi-Markov states are specified with `logrrnext`. For example, `logrrnext(age,2,3)` for the effect of age on the relative rate of transition from state 2 to state 3, relative to the rate of transition from state 2 to the first competing destination state. These parameters are not applicable to semi-Markov states with only one potential next destination state.

In general, the indices or the covariate name can be omitted to indicate that the same prior will be used for all transitions, or/and all covariates. This can be done with or without the brackets, e.g. `"logq()"` or `"logq"` are both understood.

mean	Prior mean. This is only used for the parameters that have direct normal priors, that is <code>logq</code> , <code>loghr</code> , <code>logtaf</code> , <code>logshape</code> , <code>logscale</code> , <code>loe</code> , <code>logoddsnext</code> . That is, excluding <code>time</code> , <code>q</code> and <code>hr</code> , whose priors are defined by transformations of a normal distribution.
sd	Prior standard deviation (only for parameters with direct normal priors)
median	Prior median
lower	Prior lower 95% quantile
upper	Prior upper 95% quantile

Details

In `msmbayes`, a normal prior is used for the log transition intensities (`logq`) and log hazard ratios (`loghr`). The goal of this function is to determine the mean and SD of this prior. It can be used in two ways:

- (a) directly specifying the prior mean and SD of `logq` or `loghr`
- (b) specifying prior quantiles for more interpretable transformations of these. These may include `q` (the transition intensity) or `time` (the reciprocal of the intensity, interpreted as a mean time to this transition when observing a sequence of individuals at risk of it). Or `hr` (hazard ratio)

Two quantiles (out of the median, lower or upper) should be provided. If all three are provided, then the upper quantile is ignored. These are transformed back to the scale of `logq` or `loghr`, and the unique normal prior with these quantiles is deduced.

Value

A list of class "msmprior", with components

- par (as supplied by the user)
- par_base (e.g. "logq" if "time" was provided, or "loghr" if "hr" was provided)
- covname (name of covariate effect)
- ind1, ind2 (as supplied by the user)
- mean (of log-normal prior on par_base)
- sd (of log-normal prior on par_base)

Examples

```
priors <- list(
  msmprior("logq(1,2)", median=-2, lower=-4),
  msmprior("q(2,1)", median=0.1, upper=10)
)
Q <- rbind(c(0,1),c(1,0))
mod <- msmbayes(data=infsim2, state="state", time="months", subject="subject",
  Q=Q, priors=priors, fit_method="optimize")
summary(mod)
```

n3_moment_bounds

*Bounds on normalised moments for phase-type approximations***Description**

From Bobbio et al. (Theorem 3.1)

Usage

```
n3_moment_bounds(n2, n)
```

Arguments

n2	Second normalised moment
n	Number of phases

Value

List with components lower, upper defining the bounds on the third normalised moment (n3) required for n2 and n3 to be the moments of a phase type distribution with n phases.

nphase	<i>Density, probability distribution, quantile, moment, hazard and random number generation functions for the Coxian phase-type distribution with any number of phases.</i>
--------	---

Description

Density, probability distribution, quantile, moment, hazard and random number generation functions for the Coxian phase-type distribution with any number of phases.

Usage

```
dnphase(x, prate, arate, initp = NULL, method = "expm")
pnphase(q, prate, arate, initp = NULL, method = "expm", lower.tail = TRUE)
hnphase(x, prate, arate, initp = NULL, method = "expm")
mean_nphase(prate, arate, initp = NULL)
var_nphase(prate, arate, initp = NULL)
skewness_nphase(prate, arate, initp = NULL)
ncmoment_nphase(prate, arate, i, initp = NULL)
rnphase(n, prate, arate)
qnphase(p, prate, arate, lower.tail = TRUE, log.p = FALSE, method = "expm")
```

Arguments

x	Value at which to evaluate the PDF, CDF, or hazard.
prate	Progression rates. Either a vector of length nphase-1, or a matrix with npar rows and nphase-1 columns.
arate	Absorption rates. Either a vector of length nphase, or a matrix with npar rows and nphase columns.
initp	Vector of probabilities of occupying each phase at the start of the sojourn. By default, the first phase has probability 1.
method	If "analytic" then for nphase 5 or less, an analytic solution to the matrix exponential is employed in the calculation. For nphase 6 or more, or if method="mexp" the matrix exponential is determined using numerical methods, via expm::expm().
q	Value at which to evaluate the CDF.
lower.tail	If TRUE return $P(X < x)$, else $P(X \geq x)$.
i	which moment to return from ncmoment_nphase

n	Number of random samples to generate.
p	Probability at which to evaluate the quantile
log.p	return log probability

Details

The number of phases, `nphase`, is taken from the dimensions of the object supplied as `arate`. If `arate` is a vector, then the number of phases is assumed to equal the length of this vector. If `arate` is a matrix, then the number of phases is assumed to be the number of columns.

`mean_nphase`, `var_nphase`, `skewness_nphase` and `ncmoment_nphase` return the mean, variance, skewness and general non-central moments of the distribution.

These functions work in a vectorised way, so that alternative parameter values or evaluation values `x` can be supplied. The number of alternative values is determined from the number of rows `nrep` of `arate`. Then if necessary, `prate` and `x` are replicated to match the size of `arate`.

Value

A vector of length `n` or `length(x)`.

nphase_Q	<i>Given a phase-type sojourn distribution, return the corresponding Markov intensity matrix where the last state is the absorbing state, and the the time to absorption is the sojourn distribution.</i>
----------	---

Description

Given a phase-type sojourn distribution, return the corresponding Markov intensity matrix where the last state is the absorbing state, and the the time to absorption is the sojourn distribution.

Usage

```
nphase_Q(prate, arate)
```

Arguments

prate	Progression rates. Either a vector of length <code>nphase-1</code> , or a matrix with <code>npar</code> rows and <code>nphase-1</code> columns.
arate	Absorption rates. Either a vector of length <code>nphase</code> , or a matrix with <code>npar</code> rows and <code>nphase</code> columns.

phaseapprox_pars	<i>Summarise posteriors for shape and scale parameters for the sojourn distribution in a semi-Markov msmbayes model</i>
------------------	---

Description

In models with covariates on the scale parameter, this currently only presents these parameters for covariate values of zero.

Usage

```
phaseapprox_pars(draws, log = FALSE)
```

Arguments

draws	Object returned by msmbayes .
log	Return parameters on log scale

pmatrix	<i>Transition probability matrix from an msmbayes model</i>
---------	---

Description

Transition probability matrix from an msmbayes model

Usage

```
pmatrix(
  draws,
  t = 1,
  new_data = NULL,
  states = "obs",
  drop = TRUE,
  type = "posterior"
)
```

Arguments

draws	Object returned by msmbayes .
t	prediction time or vector of prediction times
new_data	Data frame with covariate values to predict for

states	If states="obs" (or "observed") then this describes mean sojourn times in the observable states. For phase-type models this is not generally equal to the sum of the phase-specific mean sojourn times, because an individual may transition out of the state before progressing to the next phase. If states="phase" (or "true", or "latent") then for phase-type models, this describes mean sojourn times in the latent state space.
drop	Only used if there are no covariates supplied in new_data. Then if drop=TRUE this returns a nstates x nstates matrix, or if drop=FALSE this returns a 3D array with first dimension ncovs=1.
type	"posterior" to return rvar objects containing posterior samples. "mode" to return the output evaluated at the posterior mode of the basic parameters (only applicable if model was fitted with posterior mode optimisation).

Value

Array or matrix of rvar objects giving the transition probability matrix at each requested prediction time and covariate value. See [qdf](#) for notes on the rvar format.

For phase-type models, if states="obs", so that we want transition probabilities on the observable space, this returns the probability of transition to any phase of each "destination" state, for an individual who is in the first phase of each "starting" state.

See Also

[pmatrixdf](#) returns the same information in a tidy data frame format.

pmatrixdf	<i>Transition probabilities from an msmbayes model, presented as a tidy data frame</i>
-----------	--

Description

Transition probabilities from an msmbayes model, presented as a tidy data frame

Usage

```
pmatrixdf(draws, t = 1, new_data = NULL, states = "obs")
```

Arguments

draws	Object returned by msmbayes .
t	prediction time or vector of prediction times
new_data	Data frame with covariate values to predict for
states	If states="obs" (or "observed") then this describes mean sojourn times in the observable states. For phase-type models this is not generally equal to the sum of the phase-specific mean sojourn times, because an individual may transition out of the state before progressing to the next phase. If states="phase" (or "true", or "latent") then for phase-type models, this describes mean sojourn times in the latent state space.

Value

A data frame containing samples from the posterior distribution. See [qdf](#) for notes on this format and how to summarise.

For phase-type models, if `states="obs"`, so that we want transition probabilities on the observable space, this returns the probability of transition to any phase of each "destination" state, for an individual who is in the first phase of each "starting" state.

pnext

Probabilities for the next state in a multi-state model

Description

Given an individual is currently in state r , these are the probabilities that when leaving state r , the individual will move to a particular state s .

Usage

```
pnext(draws, new_data = NULL)
```

Arguments

<code>draws</code>	Object returned by msmbayes .
<code>new_data</code>	Data frame with covariate values to predict for

Details

In a Markov model, this is defined as the transition intensity from r to s divided by the sum of all transition intensities out of r .

In semi-Markov models, this quantity is a model parameter in itself. In phase-type approximation models, the parameters consist of the parameters of the sojourn distribution and the next-state probabilities, which (as in a Markov model) are assumed to be independent of the sojourn time.

As the models in `msmbayes` work in continuous time, the next-state probability is different from the transition probability. The transition probability is the probability that the individual is in state s at a specific time in the future, and can be obtained from an `msmbayes` model with the functions [pdf](#), [pmatrix](#).

qdf	<i>Transition intensities from an msmbayes model, presented as a tidy data frame</i>
-----	--

Description

Transition intensities from an msmbayes model, presented as a tidy data frame

Usage

```
qdf(draws, new_data = NULL, keep_covid = FALSE)
```

Arguments

draws	Object returned by msmbayes .
new_data	Data frame with covariate values to predict for
keep_covid	(logical) Keep the integer column covid identifying unique covariate combinations.

Value

A data frame with one row per from-state / to-state / covariate value.

Column posterior is in the rvar format of the **posterior** package, representing a sample from a posterior distribution. Use the `summary` function on the data frame to produce summary statistics such as the posterior median or mean (see [summary.msmbayes](#)).

See Also

[qmatrix](#) returns the same information in matrix format

Examples

```
qdf(infsim_model)
summary(qdf(infsim_model))
summary(qdf(infsim_model), median, ~quantile(.x, c(0.025, 0.975)))

qdf(infsim_modelc,
    new_data = data.frame(sex=c("female", "male")))
```

qmatrix

*Transition intensity matrix from an msmbayes model***Description**

Transition intensity matrix from an msmbayes model

Usage

```
qmatrix(draws, new_data = NULL, X = NULL, drop = TRUE, type = "posterior")
```

Arguments

draws	Object returned by msmbayes .
new_data	Data frame with covariate values to predict for
X	Lower-level alternative to specifying new_data, for developer use only. X is a numeric matrix formed from column-binding the covariate design matrices for each transition in turn.
drop	Only used if there are no covariates supplied in new_data. Then if drop=TRUE this returns a nstates x nstates matrix, or if drop=FALSE this returns a 3D array with first dimension ncovs=1.
type	"posterior" to return rvar objects containing posterior samples. "mode" to return the output evaluated at the posterior mode of the basic parameters (only applicable if model was fitted with posterior mode optimisation).

Value

An array or matrix of rvar objects or numbers, representing the transition intensity matrix for each new prediction data point

See Also

[qdf](#) returns the same information in a tidy data frame format

Examples

```
qmatrix(infsim_model)
summary(qmatrix(infsim_model))
summary(qmatrix(infsim_model), median, ~quantile(.x, c(0.025, 0.975)))
```

qphaseapprox	<i>Phase-type expansion of a transition intensity matrix to create a non-Markov multi-state model</i>
--------------	---

Description

Convert a multi-state model intensity matrix with one or more non-Markov states to an intensity matrix on a phase-type state space, where the non-Markov states are modelled with a phase-type approximation of a shape/scale distribution.

Usage

```
qphaseapprox(
  qmatrix,
  pastates,
  shape,
  scale = NULL,
  family = "gamma",
  nphase = NULL,
  att = FALSE
)
```

Arguments

qmatrix	Intensity matrix on the observable state space. Only the rates for transitions out of Markov states are used, and values of rates for transitions out of the non-Markov state are ignored, unless there are competing next states. In that case the relative value of the intensities are interpreted as the transition probability to each next state. These transition probabilities are multiplied by the phase transition rates of the sojourn distribution in the non-Markov state to get the transition rates from the phases to the destination state.
pastates	This indicates which states (if any) are given a Weibull or Gamma sojourn distribution approximated by a phase-type model. Ignored if nphase is supplied.
shape	shape parameter. This can be vectorised.
scale	scale parameter. This can be vectorised.
family	parametric family approximated by the phase-type distribution: "weibull" or "gamma"
nphase	Only required for models with phase-type sojourn distributions specified directly (not through pastates). nphase is a vector with one element per state, giving the number of phases per state. This element is 1 for states that do not have phase-type sojourn distributions.
att	keep attributes indicating progression and absorption states

Value

Intensity matrix on the latent state space.

rrnextdoc	<i>Effects of covariates on competing exit transitions in phase type models</i>
-----------	---

Description

Effects of covariates on competing exit transitions in phase type models

Usage

```
logrrnext(draws)
```

```
rrnext(draws)
```

Arguments

draws Object returned by [msmbayes](#).

Details

Only applicable to phase-type approximation models, specified with `pastates`.

`logrrnext` gives the Linear effect of covariates on log relative risk of transition to a competing destination state, relative to baseline destination state.

`rrnext` gives the exponential of the linear effect, interpretable as a hazard ratio. See the semi-Markov models vignette and paper for the mathematical details.

Value

A data frame containing samples from the posterior distribution. See [qdf](#) for notes on this format and how to summarise.

shape_in_bounds	<i>Test whether a shape parameter of is in the bounds required for a valid phase-type approximation</i>
-----------------	---

Description

Test whether a shape parameter of is in the bounds required for a valid phase-type approximation

Usage

```
gamma_shape_in_bounds(shape, nphase)
```

```
weibull_shape_in_bounds(shape, nphase)
```

Arguments

shape	Shape parameter or vector)
nphase	Number of phases

Details

Also verifies that the parameter satisfies Case 1 of Theorem 1 in Bobbio et al.

Value

Vector or logicals, whether each shape parameter is in the bounds require for a phase-type approximation with that number of phases.

shape_ubound	<i>Upper bound for shape parameter in moment-based phase-type approximations</i>
--------------	--

Description

Upper bound for shape parameter in moment-based phase-type approximations

Usage

shape_ubound(nphase, family)

Arguments

nphase	Number of approximating phases
family	"weibull" or "gamma"

Value

Upper bound for shape parameter

shapyscale_to_rates	<i>Determine parameters of a phase-type model that approximate a parametric shape-scale distribution</i>
---------------------	--

Description

Determine parameters of a phase-type model that approximate a parametric shape-scale distribution

Usage

```
shapyscale_to_rates(
  shape,
  scale = 1,
  family = "gamma",
  canonical = FALSE,
  nphase = 3,
  list = FALSE,
  drop = TRUE
)
```

Arguments

shape	shape parameter. This can be vectorised.
scale	scale parameter. This can be vectorised.
family	parametric family approximated by the phase-type distribution: "weibull" or "gamma"
canonical	Return the phase-type parameters in canonical form (phase 1 sojourn rate, sojourn rate increments in subsequent states, absorption probabilities). If FALSE then phase transition rates are returned.
nphase	Number of phases.
list	If TRUE then separate components are returned for progression and absorption rates. Otherwise, and by default, a vector (or matrix) is returned combining all rates. If a vector is supplied for shape or scale, the returned object (or the list components) is a matrix.
drop	If shape or scale have both have one element, and drop=FALSE, a matrix with one row is returned.

Details

The approximating phase-type distribution is one for which the first three moments are the same as those of the target distribution. See the vignettes and paper for full details.

sim_2state_smm	<i>Simulate intermittently-observed data from a semi-Markov multi-state model with two states and reversible transitions.</i>
----------------	---

Description

Either or both states can have any sojourn distribution that we know how to simulate from.

Usage

```
sim_2state_smm(
  nindiv,
  obstimes,
  rfn1 = rexp,
  pars1 = list(rate = 1),
  rfn2 = exp,
  pars2 = list(rate = 1)
)
```

Arguments

nindiv	Number of individuals.
obstimes	Observation times, common between individuals.
rfn1	Function to simulate from the sojourn distribution in state 2. By default this is the exponential.
pars1	Named list of arguments and their values to be passed to rfn1, specifying parameter values for the sojourn distribution.
rfn2	Function to simulate from the sojourn distribution in state 2.
pars2	Named list of arguments and their values to be passed to rfn2, specifying parameter values for the sojourn distribution.

soj_prob	<i>Sojourn probability in a state of a msmbayes model</i>
----------	---

Description

Sojourn probability in a state of a msmbayes model

Usage

```
soj_prob(draws, t, state, new_data = NULL, method = "analytic")
```

Arguments

draws	Object returned by msmbayes .
t	Time since state entry. A single time or a vector can be supplied.
state	State of interest (A single integer)
new_data	Data frame with covariate values to predict for
method	Only applicable to phase-type models. Method for computing the matrix exponential involved in the phase-type sojourn distribution. See pnphase .

Details

For the inverse of this function, see [soj_quantile](#).

Value

A data frame with column `posterior` giving the posterior distribution for the probability of remaining in state by time `t` since state entry, as an `rvar` object. Other columns give the time and any covariate values.

Note this is the analogue of the survival probability, or one minus the CDF of the time-to-event distribution

See [qdf](#) for notes on the `rvar` format.

soj_quantile	<i>Quantiles of the sojourn distribution in a state of a msmbayes model</i>
--------------	---

Description

Quantiles of the sojourn distribution in a state of a `msmbayes` model

Usage

```
soj_quantile(draws, p, state, new_data = NULL, method = "analytic")
```

Arguments

draws	Object returned by msmbayes .
p	Vector of probabilities at which to evaluate quantiles of the sojourn distribution
state	State of interest (A single integer)
new_data	Data frame with covariate values to predict for
method	Only applicable to phase-type models. Method for computing the matrix exponential involved in the phase-type sojourn distribution. See pnphase .

Details

For the inverse of this function, see [soj_prob](#).

Value

A data frame with column posterior giving the posterior distribution (as an rvar object) for the time in state for which the probability of remaining in this state by this time is p.

standardise_to	<i>Constructor for a standardising population used for model outputs</i>
----------------	--

Description

Standardised outputs are outputs from models with covariates, that are defined by marginalising (averaging) over covariate values in a given population, rather than being conditional on a given covariate value.

Usage

```
standardise_to(new_data)
```

```
standardize_to(new_data)
```

Arguments

new_data Data frame with covariate values to predict for

Details

Standardised outputs are produced from a Monte Carlo sample from the joint distribution of parameters θ and covariate values X , $p(X, \theta) = p(\theta|X)p(X)$, where $p(X)$ is defined by the empirical distribution of covariates in the standard population. This joint sample is obtained by concatenating samples of covariate-specific outputs.

Hence applying an output function $g()$ (such as the transition probability) to this sample produces a sample from the posterior of $\int g(\theta|X)dX$: the average transition probability (say) for a heterogeneous population.

Examples

```
nd <- data.frame(sex=c("female", "male"))

## gender-specific outputs
qdf(infsim_modelc, new_data = nd)

## averaged over men and women in the same proportions as are in `nd`
## in this case, `nd` has two rows, so we take a 50/50 average
qdf(infsim_modelc, new_data = standardise_to(nd))
```

statetable	<i>Summarise intermittently-observed multi-state data</i>
------------	---

Description

Tabulate observed transitions between states over successive observations, by from-state, to-state and (optionally) time interval length and covariate values.

Usage

```
statetable(
  data,
  state = "state",
  subject = "subject",
  time = "time",
  covariates = NULL,
  time_groups = 1,
  format = "wide"
)
```

Arguments

data	Data frame giving the observed data.
state	Character string naming the observed state variable in the data. This variable must either be an integer in 1,2,...,K, where K is the number of states, or a factor with these integers as level labels. If omitted, this is assumed to be "state".
subject	Character string naming the individual ID variable in the data. If omitted, this is assumed to be "subject".
time	Character string naming the observation time variable in the data. If omitted, this is assumed to be "time".
covariates	Vector of names of covariates to summarise counts by.
time_groups	Number of groups to summarise the time intervals by. The transitions are categorised into groups according to equally-spaced quantiles of the time interval length.
format	"long" to return one row per tostate (a pure "tidy data" format) or "wide" to return one column per tostate (like statetable.msm in msm).

Details

This is like the function `statetable.msm` in **msm**, except that it uses `msmbayes` syntax for specifying the data, it summarises the length of the time intervals between successive observations, and it returns a tidy data frame.

Warning: it is not appropriate to choose the transition structure (the `Q` argument to `msmbayes()`) on the basis of this summary. `statetable` counts transitions over a *time interval*, whereas `Q` indicates which *instantaneous* transitions are possible. The structures will not be the same. For example, in

a model with instantaneous transitions from mild to moderate illness, and moderate to severe, we might observe transitions from mild to severe over an interval of 1 year (say), but the instantaneous transition from mild to severe is impossible.

Note this is not fully tidy-friendly, as it will not work if data is grouped using dplyr.

Value

A data frame with columns fromstate, timelag and n (count of transitions), and column or columns for tostate.

summary.msmbayes	<i>Summarise basic parameter estimates from an msmbayes model</i>
------------------	---

Description

Summarise basic parameter estimates from an msmbayes model

Usage

```
## S3 method for class 'msmbayes'
summary(object, pars = NULL, ...)
```

Arguments

object	Object returned by msmbayes .
pars	Character string indicating the parameters to include in the summary. This can include: - q: transition intensities. In semi-Markov models specified with pastates these refer to the intensities of transition between the latent phases. - logq: log transition intensities - time: inverse transition intensities (mean time to event without competing risks) - mst: mean sojourn times - shape, scale: shape and/or scale for Weibull/Gamma phase-type approximations - logshape, logscale corresponding log shape or scale - pnext, logoddsnext next-state probabilities (or log odds) in phase-type approximation models - hr: hazard ratios on transition intensities, including effects on scale parameters in phase-type approximation models. - loghr: log hazard ratios - taf, logtaf: effects on scale parameters in semi-Markov phase-type approximations. - rrnext, logrrnext: effects on competing risk transition probabilities in semi-Markov phase-type approximations. - e: misclassification probabilities This defaults to whichever of c("q", "mst", "hr", "shape", "scale", "e") are included in the model.

... Further arguments passed to both `qdf`, `hr`, `loghr` and `edf`.

Value

A data frame with one row for each basic model parameter, plus rows for the mean sojourn times. The posterior distribution for the parameter is encoded in the column `posterior`, which has the `rvar` data type defined by the **posterior package**. This distribution can be summarised in any way by calling `summary` again on the data frame (see the examples).

Transition intensities, or transformations of transition intensities, are those for covariate values of zero.

Remaining parameters (in non-HMMs) are log hazard ratios for covariate effects.

The columns `prior` and `prior_string` summarise the corresponding prior distribution in two different ways. `prior` is a quasi-random sample from the prior in the `rvar` data type, and is printed as mean and standard deviation. This sample can then be used to produce any summary or plot of the prior. The string `prior_string` is a summary of this sample, showing the median and 95% equal tailed credible interval.

See Also

`qdf`, `hr`, `loghr`, `posterior::summarise_draws`

Examples

```
summary(infsim_model)
summary(summary(infsim_model))
summary(summary(infsim_model), median, ~quantile(.x, 0.025, 0.975))
```

`totlos`

Total length of stay in each state over an interval

Description

See `msm::totlos.msm()` for the theory behind the method used to calculate this. The analytic formula is used, not numerical integration.

Usage

```
totlos(
  draws,
  t,
  new_data = NULL,
  fromt = 0,
  pstart = NULL,
  discount = 0,
  states = "obs"
)
```

Arguments

draws	Object returned by msmbayes .
t	End point of the time interval over which to measure length of stay in each state
new_data	Data frame with covariate values to predict for
fromt	Starting point of the time interval, by default 0
pstart	Vector giving distribution of states at time 0
discount	Discount rate in continuous time
states	If <code>states="obs"</code> (or <code>"observed"</code>) then this describes mean sojourn times in the observable states. For phase-type models this is not generally equal to the sum of the phase-specific mean sojourn times, because an individual may transition out of the state before progressing to the next phase. If <code>states="phase"</code> (or <code>"true"</code>, or <code>"latent"</code>) then for phase-type models, this describes mean sojourn times in the latent state space.

Value

Data frame with one row for each state and covariate value, giving the expected amount of time spent in that state over the forecast interval.

Index

* **datasets**
 bigdat, 3
 example_models, 6
 illdeath_data, 6
 infsim, 7

bigdat, 3

canpars_to_rates, 4

dnphase (nphase), 28

edf, 4, 8, 44

ematrix, 5

example_models, 6

gamma_shape_in_bounds
 (shape_in_bounds), 36

hnphase (nphase), 28

hr, 8, 15, 44

hr (loghr), 8

illdeath_data, 6

infsim, 6, 7

infsim2 (infsim), 7

infsim_model (example_models), 6

infsim_modelc (example_models), 6

infsim_modelp (example_models), 6

infsim_modelpc (example_models), 6

loghr, 8, 8, 44

loglik, 8

logLik.msmbayes (loglik), 8

logoddsnext, 9

logrrnext (rrnextdoc), 36

logtaf, 8, 9, 10

mean_nphase (nphase), 28

mean_sojourn, 10

msm::totlos.msm(), 44

msmbayes, 5, 8–10, 11, 16, 30–34, 36, 40, 43, 45

msmbayes-package, 3

msmbayes_prior_sample, 15

msmbayes_priorpred_sample, 18

msmhst, 21, 24

msmhst_bardata, 23, 23

msmprior, 14, 18, 20, 21, 25

n3_moment_bounds, 27

ncmoment_nphase (nphase), 28

nphase, 28

nphase_Q, 29

pdf, 32

phaseapprox_pars, 30

pmatrix, 30, 32

pmatrixdf, 31, 31

pnext, 32

pnphase, 40

pnphase (nphase), 28

posterior::summarise_draws, 44

qdf, 5, 8, 10, 15, 31, 32, 33, 34, 36, 40, 44

qmatrix, 33, 34

qnphase (nphase), 28

qphaseapprox, 35

rates_to_canpars (canpars_to_rates), 4

rnphase (nphase), 28

rrnext (rrnextdoc), 36

rrnextdoc, 36

rstan::optimizing(), 15

rstan::sampling(), 15

rstan::vb(), 15

shape_in_bounds, 36

shape_ubound, 37

shapyscale_to_rates, 38

sim_2state_smm, 39

skewness_nphase (nphase), 28

soj_prob, [39](#), [40](#)
soj_quantile, [40](#), [40](#)
standardise_to, [41](#)
standardize_to (standardise_to), [41](#)
statetable, [42](#)
summary.msmbayes, [15](#), [33](#), [43](#)

taf, [10](#)
taf (logtaf), [9](#)
totlos, [44](#)

var_nphase (nphase), [28](#)

weibull_shape_in_bounds
 (shape_in_bounds), [36](#)