Package ‘multibridge’

December 3, 2025

Title Evaluating Multinomial Order Restrictions with Bridge Sampling

Version 1.3.0

Date 2025-12-01

Description Evaluate hypotheses concerning the distribution of multinomial proportions using bridge sampling. The bridge sampling routine is able to compute Bayes factors for hypotheses that entail inequality constraints, equality constraints, free parameters, and mixtures of all three. These hypotheses are tested against the encompassing hypothesis, that all parameters vary freely or against the null hypothesis that all category proportions are equal. For more information see Sarafoglou et al. (2020) <doi:10.31234/osf.io/bux7p>.

URL `https://github.com/asarafoglou/multibridge/`

BugReports `https://github.com/asarafoglou/multibridge/issues`

License GPL-2

Encoding UTF-8

LazyData true

Imports Brobdingnag, coda, mvtnorm, purrr, Rcpp (>= 0.12.17), magrittr, progress, Rdpack, stringr

Suggests knitr, rmarkdown, testthat

SystemRequirements GNU make, mpfr (>= 3.0.0), gmp (>= 6.2.1_1)

RoxygenNote 7.3.2

LinkingTo Rcpp

VignetteBuilder knitr

RdMacros Rdpack

Depends R (>= 3.5.0)

NeedsCompilation yes

Author Alexandra Sarafoglou [aut, cre] (ORCID: <`https://orcid.org/0000-0003-0031-685X`>),
Frederik Aust [aut] (ORCID: <`https://orcid.org/0000-0003-4900-788X`>),
Julia M. Haaf [aut] (ORCID: <`https://orcid.org/0000-0001-5122-706X`>),
Joris Goosen [aut],

Quentin F. Gronau [aut],
Maarten Marsman [aut] (ORCID: <https://orcid.org/0000-0001-5309-7502>)

Maintainer Alexandra Sarafoglou <`alexandra.sarafoglou@gmail.com`>

Repository CRAN

Date/Publication 2025-12-03 13:00:02 UTC

Contents

`.adjustUpperBoundForFreeParameters` *Adjusts Upper Bound For Free Parameters*

Description

Corrects the upper bound for current parameter. This correction only applies for parameters that are free to vary within the restriction. Then the length of the remaining stick must be based on the largest free parameter value.

Usage

```
.adjustUpperBoundForFreeParameters(
  theta_mat,
  k,
  upper,
  nr_mult_equal,
  smaller_values,
  larger_values,
  hyp_direction
)
```

Arguments

`theta_mat`	matrix with samples from truncated Dirichlet density
`k`	current parameter index
`upper`	current upper bound
`nr_mult_equal`	vector of multiplicative elements of collapsed parameters
`smaller_values`	index of parameters that are smaller than the current one
`larger_values`	index of parameters that are larger than the current one
`hyp_direction`	specifies whether the imposed inequality constrained imposes an increasing (i.e., ’smaller’) or decreasing (i.e., ’larger’) trend

Value

adjusted upper bound

`.computeLengthOfRemainingStick` *Computes Length Of Remaining Stick*

Description

When applying the probit transformation on the Dirichlet samples, this function is used as part of the stick-breaking algorithm. It computes the length of the remaining stick when the current element is broken off.

Usage

```
.computeLengthOfRemainingStick(theta_mat, k, hyp_direction)
```

Arguments

`theta_mat`	matrix with samples from truncated Dirichlet density
`k`	current parameter index
`hyp_direction`	specifies whether the imposed inequality constrained imposes an increasing (i.e., 'smaller') or decreasing (i.e., 'larger') trend

Value

stick length

`bayes_factor` *S3 method for class 'bayes_factor.bmult'*

Description

Extracts information about computed Bayes factors from object of class `bmult`

Usage

```
bayes_factor(x)
```

Arguments

`x`	object of class `bmult` as returned from `mult_bf_informed` or `binom_bf_informed` or an object of class `bmult_bridge` as returned from `mult_bf_inequality` or `binom_bf_inequality`

Value

Returns `list` with three `data.frames`. The first dataframe `bf_table` summarizes information the Bayes factor for equality and inequality constraints. The second dataframe `error_measures` contains for the overall Bayes factor the approximate relative mean-squared error `re2`, the approximate coefficient of variation `cv`, and the approximate percentage error `percentage`. The third dataframe `$bf_ineq_table` summarized information about the Bayes factor for inequality constraints, that is, the log marginal likelihood estimates for the constrained prior and posterior distribution. In addition, it contains for each independent Bayes factor the approximate relative mean-squared error `re2`

Note

In case the restricted hypothesis is tested against H_0 four `data.frames` will be returned. The fourth dataframe `$bf_eq_table` summarizes information about the Bayes factor for equality constraints compared to the encompassing hypothesis.

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=500, seed=2020)
bayes_factor(out_mult)
```

`bayes_factor.bmult` *Extracts information about computed Bayes factors from object of class* `bmult`

Description

Extracts information about computed Bayes factors from object of class `bmult`

Usage

```
## S3 method for class 'bmult'
bayes_factor(x)
```

Arguments

x	object of class `bmult` as returned from `mult_bf_informed` or `binom_bf_informed` or an object of class `bmult_bridge` as returned from `mult_bf_inequality` or `binom_bf_inequality`

Value

Returns `list` with three `data.frames`. The first dataframe `bf_table` summarizes information the Bayes factor for equality and inequality constraints. The second dataframe `error_measures` contains for the overall Bayes factor the approximate relative mean-squared error `re2`, the approximate coefficient of variation `cv`, and the approximate percentage error `percentage`. The third dataframe `$bf_ineq_table` summarized information about the Bayes factor for inequality constraints, that is, the log marginal likelihood estimates for the constrained prior and posterior distribution. In addition, it contains for each independent Bayes factor the approximate relative mean-squared error `re2`

Note

In case the restricted hypothesis is tested against H_0 four `data.frames` will be returned. The fourth dataframe `$bf_eq_table` summarizes information about the Bayes factor for equality constraints compared to the encompassing hypothesis.

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=500, seed=2020)
bayes_factor(out_mult)
```

`binom_bf_equality` *Computes Bayes Factors For Equality Constrained Binomial Parameters*

Description

Computes Bayes factor for equality constrained binomial parameters. Null hypothesis H_0 states that binomial proportions are exactly equal or exactly equal and equal to `p`. Alternative hypothesis H_e states that binomial proportions are free to vary.

Usage

```
binom_bf_equality(x, n = NULL, a, b, p = NULL)
```

Arguments

`x`	a vector of counts of successes, or a two-dimensional table (or matrix) with 2 columns, giving the counts of successes and failures, respectively
`n`	numeric. Vector of counts of trials. Must be the same length as `x`. Ignored if `x` is a matrix or a table
`a`	numeric. Vector with alpha parameters. Must be the same length as `x`. Default sets all alpha parameters to 1
`b`	numeric. Vector with beta parameters. Must be the same length as `x`. Default sets all beta parameters to 1
`p`	numeric. Hypothesized probability of success. Must be greater than 0 and less than 1. Default sets all binomial proportions exactly equal without specifying a specific value.

Details

The model assumes that the data in `x` (i.e., $x_1, ..., x_K$) are the observations of K independent binomial experiments, based on $n_1, ..., n_K$ observations. Hence, the underlying likelihood is the product of the $k = 1, ..., K$ individual binomial functions:

$$(x_1, ...x_K) \prod Binomial(N_k, \theta_k)$$

Furthermore, the model assigns a beta distribution as prior to each model parameter (i.e., underlying binomial proportions). That is:

$$\theta_k \ Beta(\alpha_k, \beta_k)$$

Value

Returns a `data.frame` containing the Bayes factors `LogBFe0`, `BFe0`, and `BF0e`

Note

The following signs can be used to encode restricted hypotheses: `"<"` and `">"` for inequality constraints, `"="` for equality constraints, `","` for free parameters, and `"&"` for independent hypotheses. The restricted hypothesis can either be a string or a character vector. For instance, the hypothesis `c("theta1 < theta2, theta3")` means

- `theta1` is smaller than both `theta2` and `theta3`
- The parameters `theta2` and `theta3` both have `theta1` as lower bound, but are not influenced by each other.

The hypothesis `c("theta1 < theta2 = theta3 & theta4 > theta5")` means that

- Two independent hypotheses are stipulated: `"theta1 < theta2 = theta3"` and `"theta4 > theta5"`

- The restrictions on the parameters `theta1`, `theta2`, and `theta3` do not influence the restrictions on the parameters `theta4` and `theta5`.
- `theta1` is smaller than `theta2` and `theta3`
- `theta2` and `theta3` are assumed to be equal
- `theta4` is larger than `theta5`

References

Damien P, Walker SG (2001). "Sampling truncated normal, beta, and gamma densities." *Journal of Computational and Graphical Statistics*, **10**, 206–215.

Gronau QF, Sarafoglou A, Matzke D, Ly A, Boehm U, Marsman M, Leslie DS, Forster JJ, Wagenmakers E, Steingroever H (2017). "A tutorial on bridge sampling." *Journal of Mathematical Psychology*, **81**, 80–97.

Frühwirth-Schnatter S (2004). "Estimating marginal likelihoods for mixture and Markov switching models using bridge sampling techniques." *The Econometrics Journal*, **7**, 143–167.

Sarafoglou A, Haaf JM, Ly A, Gronau QF, Wagenmakers EJ, Marsman M (2021). "Evaluating Multinomial Order Restrictions with Bridge Sampling." *Psychological Methods*.

See Also

Other functions to evaluate informed hypotheses: `binom_bf_inequality()`, `binom_bf_informed()`, `mult_bf_equality()`, `mult_bf_inequality()`, `mult_bf_informed()`

Examples

```
data(journals)
x <- journals$errors
n <- journals$nr_NHST
a <- rep(1, nrow(journals))
b <- rep(1, nrow(journals))
binom_bf_equality(x=x, n=n, a=a, b=b)
```

`binom_bf_inequality` *Computes Bayes Factors For Inequality Constrained Independent Binomial Parameters*

Description

Computes Bayes factor for inequality constrained binomial parameters using a bridge sampling routine. Restricted hypothesis H_r states that binomial proportions follow a particular trend. Alternative hypothesis H_e states that binomial proportions are free to vary.

Usage

```
binom_bf_inequality(
  samples = NULL,
  restrictions = NULL,
  x = NULL,
  n = NULL,
  Hr = NULL,
  a = rep(1, ncol(samples)),
  b = rep(1, ncol(samples)),
  factor_levels = NULL,
  prior = FALSE,
  index = 1,
  maxiter = 1000,
  seed = NULL,
  niter = 5000,
  nburnin = niter * 0.05
)
```

Arguments

`samples`	matrix of dimension (`nsamples` x `nparams`) with samples from truncated Dirichlet density
`restrictions`	`list` of class `bmult_rl` or of class `bmult_rl_ineq` as returned from `generate_restriction_list` that encodes inequality constraints for each independent restriction
`x`	a vector of counts of successes, or a two-dimensional table (or matrix) with 2 columns, giving the counts of successes and failures, respectively
`n`	numeric. Vector of counts of trials. Must be the same length as `x`. Ignored if `x` is a matrix or a table
`Hr`	string or character. Encodes the user specified informed hypothesis. Use either specified `factor_levels` or indices to refer to parameters. See “Note” section for details on how to formulate informed hypotheses
`a`	numeric. Vector with alpha parameters. Must be the same length as `x`. Default sets all alpha parameters to 1
`b`	numeric. Vector with beta parameters. Must be the same length as `x`. Default sets all beta parameters to 1
`factor_levels`	character. Vector with category names. Must be the same length as `x`
`prior`	logical. If `TRUE` the function will ignore the data and evaluate only the prior distribution
`index`	numeric. Index of current restriction. Default is 1
`maxiter`	numeric. Maximum number of iterations for the iterative updating scheme used in the bridge sampling routine. Default is 1,000 to avoid infinite loops
`seed`	numeric. Sets the seed for reproducible pseudo-random number generation
`niter`	numeric. Vector with number of samples to be drawn from truncated distribution

`nburnin`	numeric. A single value specifying the number of burn-in samples when drawing from the truncated distribution. Minimum number of burn-in samples is 10. Default is 5% of the number of samples. Burn-in samples are removed automatically after the sampling.

Details

The model assumes that the data in `x` (i.e., $x_1, ..., x_K$) are the observations of K independent binomial experiments, based on $n_1, ..., n_K$ observations. Hence, the underlying likelihood is the product of the $k = 1, ..., K$ individual binomial functions:

$$(x_1, ...x_K) \prod Binomial(N_k, \theta_k)$$

Furthermore, the model assigns a beta distribution as prior to each model parameter (i.e., underlying binomial proportions). That is:

$$\theta_k \; Beta(\alpha_k, \beta_k)$$

Value

List consisting of the following elements:

`$eval`
- `q11`: log prior or posterior evaluations for prior or posterior samples
- `q12`: log proposal evaluations for prior or posterior samples
- `q21`: log prior or posterior evaluations for samples from proposal
- `q22`: log proposal evaluations for samples from proposal

`$niter` number of iterations of the iterative updating scheme

`$logml` estimate of log marginal likelihood

`$hyp` evaluated inequality constrained hypothesis

`$error_measures`
- `re2`: the approximate relative mean-squared error for the marginal likelihood estimate
- `cv`: the approximate coefficient of variation for the marginal likelihood estimate (assumes that bridge estimate is unbiased)
- `percentage`: the approximate percentage error of the marginal likelihood estimate

Note

The following signs can be used to encode restricted hypotheses: `"<"` and `">"` for inequality constraints, `"="` for equality constraints, `","` for free parameters, and `"&"` for independent hypotheses. The restricted hypothesis can either be a string or a character vector. For instance, the hypothesis `c("theta1 < theta2, theta3")` means

- `theta1` is smaller than both `theta2` and `theta3`
- The parameters `theta2` and `theta3` both have `theta1` as lower bound, but are not influenced by each other.

The hypothesis `c("theta1 < theta2 = theta3 & theta4 > theta5")` means that

- Two independent hypotheses are stipulated: `"theta1 < theta2 = theta3"` and `"theta4 > theta5"`

- The restrictions on the parameters `theta1`, `theta2`, and `theta3` do not influence the restrictions on the parameters `theta4` and `theta5`.
- `theta1` is smaller than `theta2` and `theta3`
- `theta2` and `theta3` are assumed to be equal
- `theta4` is larger than `theta5`

References

Gronau QF, Sarafoglou A, Matzke D, Ly A, Boehm U, Marsman M, Leslie DS, Forster JJ, Wagenmakers E, Steingroever H (2017). "A tutorial on bridge sampling." *Journal of Mathematical Psychology*, **81**, 80–97.

Sarafoglou A, Haaf JM, Ly A, Gronau QF, Wagenmakers EJ, Marsman M (2021). "Evaluating Multinomial Order Restrictions with Bridge Sampling." *Psychological Methods*.

See Also

`generate_restriction_list`

Other functions to evaluate informed hypotheses: `binom_bf_equality()`, `binom_bf_informed()`, `mult_bf_equality()`, `mult_bf_inequality()`, `mult_bf_informed()`

Examples

```
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)

# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

results_prior  <- binom_bf_inequality(Hr=Hr, a=a, b=b, niter = 100,
factor_levels=factor_levels, prior=TRUE, seed = 2020)
# corresponds to
cbind(exp(results_prior$logml), 1/factorial(4))

# alternative - if you have samples and a restriction list
inequalities  <- generate_restriction_list(Hr=Hr, a=a,b=b,
factor_levels=factor_levels)$inequality_constraints
prior_samples <- binom_tsampling(inequalities, niter = 500,
prior=TRUE, seed = 2020)
results_prior <- binom_bf_inequality(prior_samples, inequalities, niter = 100,
seed=2020)
cbind(exp(results_prior$logml), 1/factorial(4))
```

`binom_bf_informed` *Evaluates Informed Hypotheses on Multiple Binomial Parameters*

Description

Evaluates informed hypotheses on multiple binomial parameters. These hypotheses can contain (a mixture of) inequality constraints, equality constraints, and free parameters. Informed hypothesis H_r states that binomial proportions obey a particular constraint. H_r can be tested against the encompassing hypothesis H_e or the null hypothesis H_0. Encompassing hypothesis H_e states that binomial proportions are free to vary. Null hypothesis H_0 states that category proportions are exactly equal.

Usage

```
binom_bf_informed(
  x,
  n = NULL,
  Hr,
  a,
  b,
  factor_levels = NULL,
  cred_level = 0.95,
  niter = 5000,
  bf_type = "LogBFer",
  seed = NULL,
  maxiter = 1000,
  nburnin = niter * 0.05
)
```

Arguments

`x`	a vector of counts of successes, or a two-dimensional table (or matrix) with 2 columns, giving the counts of successes and failures, respectively
`n`	numeric. Vector of counts of trials. Must be the same length as `x`. Ignored if `x` is a matrix or a table
`Hr`	string or character. Encodes the user specified informed hypothesis. Use either specified `factor_levels` or indices to refer to parameters. See “Note” section for details on how to formulate informed hypotheses
`a`	numeric. Vector with alpha parameters. Must be the same length as `x`. Default sets all alpha parameters to 1
`b`	numeric. Vector with beta parameters. Must be the same length as `x`. Default sets all beta parameters to 1
`factor_levels`	character. Vector with category names. Must be the same length as `x`
`cred_level`	numeric. Credible interval for the posterior point estimates. Must be a single number between 0 and 1

`niter`	numeric. Vector with number of samples to be drawn from truncated distribution
`bf_type`	character. The Bayes factor type. When the informed hypothesis is compared to the encompassing hypothesis, the Bayes factor type can be `LogBFer`, `BFer`, or `BFre`. When the informed hypothesis is compared to the null hypothesis, the Bayes factor type can be `LogBFr0`, `BF0r`, or `BFr0`. Default is `LogBFer`
`seed`	numeric. Sets the seed for reproducible pseudo-random number generation
`maxiter`	numeric. Maximum number of iterations for the iterative updating scheme used in the bridge sampling routine. Default is 1,000 to avoid infinite loops
`nburnin`	numeric. A single value specifying the number of burn-in samples when drawing from the truncated distribution. Minimum number of burn-in samples is 10. Default is 5% of the number of samples. Burn-in samples are removed automatically after the sampling.

Details

The model assumes that the data in `x` (i.e., $x_1, ..., x_K$) are the observations of K independent binomial experiments, based on $n_1, ..., n_K$ observations. Hence, the underlying likelihood is the product of the $k = 1, ..., K$ individual binomial functions:

$$(x_1, ...x_K) \prod Binomial(N_k, \theta_k)$$

Furthermore, the model assigns a beta distribution as prior to each model parameter (i.e., underlying binomial proportions). That is:

$$\theta_k \; Beta(\alpha_k, \beta_k)$$

Value

List consisting of the following elements

`$bf_list` gives an overview of the Bayes factor analysis:

- `bf_type`: string. Contains Bayes factor type as specified by the user
- `bf`: data.frame. Contains Bayes factors for all Bayes factor types
- `error_measures`: data.frame. Contains for the overall Bayes factor the approximate relative mean-squared error `re2`, the approximate coefficient of variation `cv`, and the approximate percentage error `percentage`
- `logBFe_equalities`: data.frame. Lists the log Bayes factors for all independent equality constrained hypotheses
- `logBFe_inequalities`: data.frame. Lists the log Bayes factor for all independent inequality constrained hypotheses

`$cred_level` numeric. User specified credible interval

`$restrictions` list that encodes informed hypothesis for each independent restriction:

- `full_model`: list containing the hypothesis, parameter names, data and prior specifications for the full model.
- `equality_constraints`: list containing the hypothesis, parameter names, data and prior specifications for each equality constrained hypothesis.

- `inequality_constraints`: list containing the hypothesis, parameter names, data and prior specifications for each inequality constrained hypothesis. In addition, in `nr_mult_equal` and `nr_mult_free` encodes which and how many parameters are equality constraint or free, in `boundaries` includes the boundaries of each parameter, in `nineq_per_hyp` states the number of inequality constraint parameters per independent inequality constrained hypothesis, and in `direction` states the direction of the inequality constraint.

`$bridge_output` list containing output from bridge sampling function:

- `eval`: list containing the log prior or posterior evaluations (`q11`) and the log proposal evaluations (`q12`) for the prior or posterior samples, as well as the log prior or posterior evaluations (`q21`) and the log proposal evaluations (`q22`) for the samples from the proposal distribution
- `niter`: number of iterations of the iterative updating scheme
- `logml`: estimate of log marginal likelihood
- `hyp`: evaluated inequality constrained hypothesis
- `error_measures`: list containing in `re2` the approximate relative mean-squared error for the marginal likelihood estimate, in `cv` the approximate coefficient of variation for the marginal likelihood estimate (assumes that bridge estimate is unbiased), and in `percentage` the approximate percentage error of the marginal likelihood estimate

`$samples` list containing a list for prior samples and a list of posterior samples from truncated distributions which were used to evaluate inequality constraints. Prior and posterior samples of independent inequality constraints are again saved in separate lists. Samples are stored as matrix of dimension `nsamples x nparams`.

Note

The following signs can be used to encode restricted hypotheses: `"<"` and `">"` for inequality constraints, `"="` for equality constraints, `","` for free parameters, and `"&"` for independent hypotheses. The restricted hypothesis can either be a string or a character vector. For instance, the hypothesis `c("theta1 < theta2, theta3")` means

- `theta1` is smaller than both `theta2` and `theta3`
- The parameters `theta2` and `theta3` both have `theta1` as lower bound, but are not influenced by each other.

The hypothesis `c("theta1 < theta2 = theta3 & theta4 > theta5")` means that

- Two independent hypotheses are stipulated: `"theta1 < theta2 = theta3"` and `"theta4 > theta5"`
- The restrictions on the parameters `theta1`, `theta2`, and `theta3` do not influence the restrictions on the parameters `theta4` and `theta5`.
- `theta1` is smaller than `theta2` and `theta3`
- `theta2` and `theta3` are assumed to be equal
- `theta4` is larger than `theta5`

References

Damien P, Walker SG (2001). “Sampling truncated normal, beta, and gamma densities.” *Journal of Computational and Graphical Statistics*, **10**, 206–215.

Gronau QF, Sarafoglou A, Matzke D, Ly A, Boehm U, Marsman M, Leslie DS, Forster JJ, Wagenmakers E, Steingroever H (2017). “A tutorial on bridge sampling.” *Journal of Mathematical Psychology*, **81**, 80–97.

Frühwirth-Schnatter S (2004). “Estimating marginal likelihoods for mixture and Markov switching models using bridge sampling techniques.” *The Econometrics Journal*, **7**, 143–167.

Sarafoglou A, Haaf JM, Ly A, Gronau QF, Wagenmakers EJ, Marsman M (2021). “Evaluating Multinomial Order Restrictions with Bridge Sampling.” *Psychological Methods*.

See Also

Other functions to evaluate informed hypotheses: `binom_bf_equality()`, `binom_bf_inequality()`, `mult_bf_equality()`, `mult_bf_inequality()`, `mult_bf_informed()`

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('binom1', 'binom2', 'binom3', 'binom4')
Hr            <- c('binom1', '<',  'binom2', '<', 'binom3', '<', 'binom4')
output_total  <- binom_bf_informed(x, n, Hr, a, b, niter=100, factor_levels, seed=2020)
```

`binom_tsampling` *Samples From Truncated Beta Densities*

Description

Based on specified inequality constraints, samples from truncated prior or posterior beta densities.

Usage

```
binom_tsampling(
  inequalities,
  index = 1,
  niter = 10000,
  prior = FALSE,
  nburnin = niter * 0.05,
  seed = NULL
)
```

Arguments

`inequalities`	list that contains inequality constraints for each independent inequality constrained hypotheses. The list is created in the `generate_restriction_list` function
`index`	numeric. If multiple independent inequality constraints are specified, this index determines for which inequality constraint samples should be drawn. Must be a single value. Default is 1
`niter`	numeric. A single value specifying the number of samples. Default is set to $10,000$
`prior`	logical. If `TRUE` ignores the data that are encoded in `inequalities` and thus samples from the prior distribution. Default is `FALSE`.
`nburnin`	numeric. A single value specifying the number of burn-in samples when drawing from the truncated distribution. Minimum number of burn-in samples is 10. Default is 5% of the number of samples. Burn-in samples are removed automatically after the sampling.
`seed`	numeric. Sets the seed for reproducible pseudo-random number generation

Details

The model assumes that the data in `x` (i.e., $x_1, ..., x_K$) are the observations of K independent binomial experiments, based on $n_1, ..., n_K$ observations. Hence, the underlying likelihood is the product of the $k = 1, ..., K$ individual binomial functions:

$$(x_1, ...x_K) \prod Binomial(N_k, \theta_k)$$

Furthermore, the model assigns a beta distribution as prior to each model parameter (i.e., underlying binomial proportions). That is:

$$\theta_k \; Beta(\alpha_k, \beta_k)$$

Value

matrix of dimension `niter * nsamples` containing samples from truncated beta distributions.

Note

When equality constraints are specified in the restricted hypothesis, this function samples from the conditional Beta distributions given that the equality constraints hold.

Only inequality constrained parameters are sampled. Free parameters or parameters that are exclusively equality constrained will be ignored.

References

Damien P, Walker SG (2001). “Sampling truncated normal, beta, and gamma densities.” *Journal of Computational and Graphical Statistics*, **10**, 206–215.

See Also

`generate_restriction_list`

Examples

```
x <- c(200, 130, 40, 10)
n <- c(200, 200, 200, 200)
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
factor_levels <- c('binom1', 'binom2', 'binom3', 'binom4')
Hr <- c('binom1 > binom2 > binom3 > binom4')

# generate restriction list
inequalities <- generate_restriction_list(x=x, n=n, Hr=Hr, a=a, b=b,
factor_levels=factor_levels)$inequality_constraints

# sample from prior distribution
prior_samples <- binom_tsampling(inequalities, niter = 500,
prior=TRUE)
# sample from posterior distribution
post_samples <- binom_tsampling(inequalities, niter = 500)
```

`bridge_output` *S3 method for class* `bridge_output.bmult`

Description

Extracts bridge sampling output from object of class `bmult`

Usage

```
bridge_output(x)
```

Arguments

`x`	object of class `bmult` as returned from `mult_bf_informed` or `binom_bf_informed`

Value

Extracts output related to the bridge sampling routine. The output contains the following elements::

`$eval`
- `q11`: log prior or posterior evaluations for prior or posterior samples
- `q12`: log proposal evaluations for prior or posterior samples
- `q21`: log prior or posterior evaluations for samples from proposal
- `q22`: log proposal evaluations for samples from proposal

`$niter` number of iterations of the iterative updating scheme

`$logml` estimate of log marginal likelihood

`$hyp` evaluated inequality constrained hypothesis

`$error_measures`
- `re2`: the approximate relative mean-squared error for the marginal likelihood estimate
- `cv`: the approximate coefficient of variation for the marginal likelihood estimate (assumes that bridge estimate is unbiased)
- `percentage`: the approximate percentage error of the marginal likelihood estimate

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=1e3, seed=2020)
bridge_output <- bridge_output(out_mult)
```

`bridge_output.bmult` *Extracts bridge sampling output from object of class* `bmult`

Description

Extracts restriction list from an object of class `bmult`

Usage

```
## S3 method for class 'bmult'
bridge_output(x)
```

Arguments

x	object of class `bmult` as returned from `mult_bf_informed` or `binom_bf_informed`

Value

Extracts restriction list and associated hypothesis from an object of class `bmult`

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
```

```
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=1e3, seed=2020)
restriction_list <- restriction_list(out_mult)
```

`generate_restriction_list` *Creates Restriction List Based On User Specified Informed Hypothesis*

Description

Encodes the user specified informed hypothesis. It creates a separate restriction list for the full model, and all independent equality and inequality constraints. The returned list features relevant information for the transformation and sampling of the model parameters, such as information about the upper and lower bound for each parameter, and the indexes of equality constrained and free parameters.

Usage

```
generate_restriction_list(x = NULL, n = NULL, Hr, a, b = NULL, factor_levels)
```

Arguments

`x`	a vector with data (for multinomial models) or a vector of counts of successes, or a two-dimensional table (or matrix) with 2 columns, giving the counts of successes and failures, respectively (for binomial models).
`n`	numeric. Vector of counts of trials. Must be the same length as `x`. Ignored if `x` is a matrix or a table
`Hr`	string or character. Encodes the user specified informed hypothesis. Use either specified `factor_levels` or indices to refer to parameters. See “Note” section for details on how to formulate informed hypotheses
`a`	numeric. Vector with concentration parameters of Dirichlet distribution (for multinomial models) or alpha parameters for independent beta distributions (for binomial models). Default sets all parameters to 1
`b`	numeric. Vector with beta parameters. Must be the same length as `x`. Default sets all beta parameters to 1
`factor_levels`	character. Vector with category names. Must be the same length as `x`

Details

The restriction list can be created for both binomial and multinomial models. If multinomial models are specified, the arguments `b` and `n` should be left empty and `x` should not be a table or matrix.

Value

Restriction list containing the following elements:

`$full_model`
- `hyp`: character. Vector containing the informed hypothesis as specified by the user
- `parameters_full`: character. Vector containing the names for each constrained parameter
- `alpha_full`: numeric. Vector containing the concentration parameters of the Dirichlet distribution (when evaluating ordered multinomial parameters) or alpha parameters of the beta distribution (when evaluating ordered binomial parameters)
- `beta_full`: numeric. Vector containing the values of beta parameters of the beta distribution (when evaluating ordered binomial parameters)
- `counts_full`: numeric. Vector containing data values (when evaluating multinomial parameters), or number of successes (when evaluating ordered binomial parameters)
- `total_full`: numeric. Vector containing the number of observations (when evaluating ordered binomial parameters, that is, number of successes and failures)

`$equality_constraints`
- `hyp`: list. Contains all independent equality constrained hypotheses
- `parameters_equality`: character. Vector containing the names for each equality constrained parameter.
- `equality_hypotheses`: list. Contains the indexes of each equality constrained parameter. Note that these indices are based on the vector of all factor levels
- `alpha_equalities`: list. Contains the concentration parameters for equality constrained hypotheses (when evaluating multinomial parameters) or alpha parameters of the beta distribution (when evaluating ordered binomial parameters).
- `beta_equalities`: list. Contains the values of beta parameters of the beta distribution (when evaluating ordered binomial parameters)
- `counts_equalities`: list. Contains data values (when evaluating multinomial parameters), or number of successes (when evaluating ordered binomial parameters) of each equality constrained parameter
- `total_equalitiesl`: list. Contains the number of observations of each equality constrained parameter (when evaluating ordered binomial parameters, that is, number of successes and failures)

`$inequality_constraints`
- `hyp`: list. Contains all independent inequality constrained hypotheses
- `parameters_inequality`: list. Contains the names for each inequality constrained parameter
- `inequality_hypotheses`: list. Contains the indices of each inequality constrained parameter
- `alpha_inequalities`: list. Contains for inequality constrained hypotheses the concentration parameters of the Dirichlet distribution (when evaluating ordered multinomial parameters) or alpha parameters of the beta distribution (when evaluating ordered binomial parameters).
- `beta_inequalities`: list. Contains for inequality constrained hypotheses the values of beta parameters of the beta distribution (when evaluating ordered binomial parameters).

- `counts_inequalities`: list. Contains for inequality constrained parameter data values (when evaluating multinomial parameters), or number of successes (when evaluating ordered binomial parameters).
- `total_inequalities`: list. Contains for each inequality constrained parameter the number of observations (when evaluating ordered binomial parameters, that is, number of successes and failures).
- `boundaries`: list that lists for each inequality constrained parameter the index of parameters that serve as its upper and lower bounds. Note that these indices refer to the collapsed categories (i.e., categories after conditioning for equality constraints). If a lower or upper bound is missing, for instance because the current parameter is set to be the smallest or the largest, the bounds take the value `int(0)`.
- `nr_mult_equal`: list. Contains multiplicative elements of collapsed categories
- `nr_mult_free`: list. Contains multiplicative elements of free parameters
- `mult_equal`: list. Contains for each lower and upper bound of each inequality constrained parameter necessary multiplicative elements to recreate the implied order restriction, even for collapsed parameter values. If there is no upper or lower bound, the multiplicative element will be 0.
- `nineq_per_hyp`: numeric. Vector containing the total number of inequality constrained parameters for each independent inequality constrained hypotheses.
- `direction`: character. Vector containing the direction for each independent inequality constrained hypothesis. Takes the values `smaller` or `larger`.

Note

The following signs can be used to encode restricted hypotheses: `"<"` and `">"` for inequality constraints, `"="` for equality constraints, `","` for free parameters, and `"&"` for independent hypotheses. The restricted hypothesis can either be a string or a character vector. For instance, the hypothesis `c("theta1 < theta2, theta3")` means

- `theta1` is smaller than both `theta2` and `theta3`
- The parameters `theta2` and `theta3` both have `theta1` as lower bound, but are not influenced by each other.

The hypothesis `c("theta1 < theta2 = theta3 & theta4 > theta5")` means that

- Two independent hypotheses are stipulated: `"theta1 < theta2 = theta3"` and `"theta4 > theta5"`
- The restrictions on the parameters `theta1`, `theta2`, and `theta3` do not influence the restrictions on the parameters `theta4` and `theta5`.
- `theta1` is smaller than `theta2` and `theta3`
- `theta2` and `theta3` are assumed to be equal
- `theta4` is larger than `theta5`

Examples

```
# Restriction list for ordered multinomial
x <- c(1, 4, 1, 10)
a <- c(1, 1, 1, 1)
```

```
factor_levels <- c('mult1', 'mult2', 'mult3', 'mult4')
Hr <- c('mult2 > mult1 , mult3 = mult4')
restrictions <- generate_restriction_list(x=x, Hr=Hr, a=a,
factor_levels=factor_levels)
```

`journals` *Prevalence of Statistical Reporting Errors*

Description

This data set, "journals" provides a summary of statistical reporting errors (i.e., inconsistencies between reported test statistic and reported p-value) of 16,695 research articles reporting results from null hypothesis significance testing (NHST). The selected articles were published in eight major journals in psychology between 1985 to 2013:

- *Developmental Psychology* (DP)
- *Frontiers in Psychology* (FP)
- *Journal of Applied Psychology* (JAP)
- *Journal of Consulting and Clinical Psychology* (JCCP)
- *Journal of Experimental Psychology: General* (JEPG)
- *Journal of Personality and Social Psychology* (JPSP)
- *Public Library of Science* (PLoS)
- *Psychological Science* (PS)

In total, Nuijten et al. (2016) recomputed 258,105 p-values with the `R` software package `statcheck` which extracts statistics from articles and recomputes the p-values. The anonymized dataset and the data documentation was openly available on the Open Science Framework (`https://osf.io/d3ukb/`; `https://osf.io/c6ap2/`).

Usage

```
data(journals)
```

Format

A `data.frame` with 8 rows and 14 variables:

Variable Name	Description
`journal`	The journal name a research article was published in.
`articles_downloaded`	The number of articles downloaded per journal.
`articles_with_NHST`	The number of articles with NHST results.
`perc_articles_with_NHST`	The percentage of all downloaded articles that had NHST results.
`nr_NHST`	The total number of NHST results.
`mean_nr_NHST_per_article_with_NHST`	The mean number of NHST results per article that had at least one NHST res
`mean_nr_NHST_per_article_all_included`	The mean number of NHST results in all downloaded articles.
`errors`	The total number of errors.

`dec_errors`	The total number of decision errors (i.e., an error that may have changed the s
`perc_errors`	The percentage of all results that was an error.
`perc_dec_errors`	The percentage of all results that was a decision error.
`perc_articles_with_errors`	The percentage of all articles that had at least one error.
`perc_articles_with_dec_errors`	The percentage of all articles that had at least one error.
`APAfactor`	APA factor: number of detected NHST results / total number of detected p va

References

Nuijten MB, Hartgerink CH, van Assen MA, Epskamp S, Wicherts JM (2016). “The prevalence of statistical reporting errors in psychology (1985–2013).” *Behavior Research Methods*, **48**, 1205–1226.

Examples

```
data(journals)
# Prior specification
# We assign a uniform Beta distribution on each binomial probability
a <- rep(1, 8)
b <- rep(1, 8)

x <- journals$errors
n  <- journals$nr_NHST
factor_levels <- levels(journals$journal)

# restricted hypothesis
Hr1 <- c('JAP , PS , JCCP , PLOS , DP , FP , JEPG < JPSP')
out <- binom_bf_informed(x=x, n=n, Hr=Hr1, a=a, b=b,
factor_levels=factor_levels, niter = 100)

summary(out)
```

`lifestresses` *Memory of Life Stresses*

Description

This data set, "lifestresses", provides the number of reported life stresses (summed across participants) that occurred in specific months prior to an interview. This data set contains the subset of 147 participants who reported one negative life event over the time span of 18 months prior to an interview. Description taken from the JASP (2020) data library.

Usage

```
data(lifestresses)
```

Format

A `data.frame` with 18 rows and 3 variables:

- `month` The month in which participants reported a stressful life event.
- `stress.freq` The number of participants who reported a life stress in the particular month prior to an interview.
- `stress.percentage` The percentage of participants who reported a life stress in the particular month prior to an interview.

References

Haberman SJ (1978). *Analysis of qualitative data: Introductory topics*, volume 1. Academic Press.

JASP Team (2022). "JASP (Version 0.16.3.0) [Computer software]." https://jasp-stats.org/.

Sarafoglou A, Haaf JM, Ly A, Gronau QF, Wagenmakers EJ, Marsman M (2021). "Evaluating Multinomial Order Restrictions with Bridge Sampling." *Psychological Methods*.

Uhlenhuth EH, Haberman SJ, Balter MD, Lipman RS (1977). "Remembering life events." In *The origins and course of psychopathology*, 117–134. Springer Verlag.

Examples

```
data(lifestresses)
# Prior specification
# We assign a uniform Dirichlet distribution, that is, we set all
# concentration parameters to 1
a             <- rep(1, 18)
x             <- lifestresses$stress.freq
factor_levels <- lifestresses$month
# Test the following restricted Hypothesis:
# Hr: month1 > month2 > ... > month18
Hr            <- paste0(1:18, collapse=">"); Hr
out  <- mult_bf_informed(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=100, bf_type = 'BFre', seed = 4)
m1 <- summary(out)
m1
```

`mult_bf_equality` *Computes Bayes Factors For Equality Constrained Multinomial Parameters*

Description

Computes Bayes factor for equality constrained multinomial parameters using the standard Bayesian multinomial test. Null hypothesis H_0 states that category proportions are exactly equal to those specified in `p`. Alternative hypothesis H_e states that category proportions are free to vary.

Usage

```
mult_bf_equality(x, a, p = rep(1/length(a), length(a)))
```

Arguments

`x`	numeric. Vector with data
`a`	numeric. Vector with concentration parameters of Dirichlet distribution. Must be the same length as `x`. Default sets all concentration parameters to 1
`p`	numeric. A vector of probabilities of the same length as `x`. Its elements must be greater than 0 and less than 1. Default is 1/K

Details

The model assumes that data follow a multinomial distribution and assigns a Dirichlet distribution as prior for the model parameters (i.e., underlying category proportions). That is:

$$x\ Multinomial(N, \theta)$$

$$\theta\ Dirichlet(\alpha)$$

Value

Returns a `data.frame` containing the Bayes factors `LogBFe0`, `BFe0`, and `BF0e`

Note

The following signs can be used to encode restricted hypotheses: `"<"` and `">"` for inequality constraints, `"="` for equality constraints, `","` for free parameters, and `"&"` for independent hypotheses. The restricted hypothesis can either be a string or a character vector. For instance, the hypothesis `c("theta1 < theta2, theta3")` means

- `theta1` is smaller than both `theta2` and `theta3`
- The parameters `theta2` and `theta3` both have `theta1` as lower bound, but are not influenced by each other.

The hypothesis `c("theta1 < theta2 = theta3 & theta4 > theta5")` means that

- Two independent hypotheses are stipulated: `"theta1 < theta2 = theta3"` and `"theta4 > theta5"`
- The restrictions on the parameters `theta1`, `theta2`, and `theta3` do not influence the restrictions on the parameters `theta4` and `theta5`.
- `theta1` is smaller than `theta2` and `theta3`
- `theta2` and `theta3` are assumed to be equal
- `theta4` is larger than `theta5`

References

Damien P, Walker SG (2001). “Sampling truncated normal, beta, and gamma densities.” *Journal of Computational and Graphical Statistics*, **10**, 206–215.

Gronau QF, Sarafoglou A, Matzke D, Ly A, Boehm U, Marsman M, Leslie DS, Forster JJ, Wagenmakers E, Steingroever H (2017). “A tutorial on bridge sampling.” *Journal of Mathematical Psychology*, **81**, 80–97.

Frühwirth-Schnatter S (2004). “Estimating marginal likelihoods for mixture and Markov switching models using bridge sampling techniques.” *The Econometrics Journal*, **7**, 143–167.

Sarafoglou A, Haaf JM, Ly A, Gronau QF, Wagenmakers EJ, Marsman M (2021). “Evaluating Multinomial Order Restrictions with Bridge Sampling.” *Psychological Methods*.

See Also

Other functions to evaluate informed hypotheses: `binom_bf_equality()`, `binom_bf_inequality()`, `binom_bf_informed()`, `mult_bf_inequality()`, `mult_bf_informed()`

Examples

```
data(lifestresses)
x <- lifestresses$stress.freq
a <- rep(1, nrow(lifestresses))
mult_bf_equality(x=x, a=a)
```

`mult_bf_inequality` *Computes Bayes Factors For Inequality Constrained Multinomial Parameters*

Description

Computes Bayes factor for inequality constrained multinomial parameters using a bridge sampling routine. Restricted hypothesis H_r states that category proportions follow a particular trend. Alternative hypothesis H_e states that category proportions are free to vary.

Usage

```
mult_bf_inequality(
  samples = NULL,
  restrictions = NULL,
  x = NULL,
  Hr = NULL,
  a = rep(1, ncol(samples)),
  factor_levels = NULL,
  prior = FALSE,
  index = 1,
  maxiter = 1000,
  seed = NULL,
  niter = 5000,
  nburnin = niter * 0.05
)
```

Arguments

`samples`	matrix of dimension `nsamples` x `nparams` with samples from truncated Dirichlet density
`restrictions`	`list` of class `bmult_rl` or of class `bmult_rl_ineq` as returned from `generate_restriction_list` that encodes inequality constraints for each independent restriction
`x`	numeric. Vector with data
`Hr`	string or character. Encodes the user specified informed hypothesis. Use either specified `factor_levels` or indices to refer to parameters. See "Note" section for details on how to formulate informed hypotheses
`a`	numeric. Vector with concentration parameters of Dirichlet distribution. Must be the same length as `x`. Default sets all concentration parameters to 1
`factor_levels`	character. Vector with category names. Must be the same length as `x`
`prior`	logical. If `TRUE` the function will ignore the data and evaluate only the prior distribution
`index`	numeric. Index of current restriction. Default is 1
`maxiter`	numeric. Maximum number of iterations for the iterative updating scheme used in the bridge sampling routine. Default is 1,000 to avoid infinite loops
`seed`	numeric. Sets the seed for reproducible pseudo-random number generation
`niter`	numeric. Vector with number of samples to be drawn from truncated distribution
`nburnin`	numeric. A single value specifying the number of burn-in samples when drawing from the truncated distribution. Minimum number of burn-in samples is 10. Default is 5% of the number of samples. Burn-in samples are removed automatically after the sampling.

Details

The model assumes that data follow a multinomial distribution and assigns a Dirichlet distribution as prior for the model parameters (i.e., underlying category proportions). That is:

$$x \; Multinomial(N, \theta)$$

$$\theta \; Dirichlet(\alpha)$$

Value

List consisting of the following elements:

`$eval`
- `q11`: log prior or posterior evaluations for prior or posterior samples
- `q12`: log proposal evaluations for prior or posterior samples
- `q21`: log prior or posterior evaluations for samples from proposal
- `q22`: log proposal evaluations for samples from proposal

`$niter` number of iterations of the iterative updating scheme

`$logml` estimate of log marginal likelihood

`$hyp` evaluated inequality constrained hypothesis

`$error_measures`
- `re2`: the approximate relative mean-squared error for the marginal likelihood estimate
- `cv`: the approximate coefficient of variation for the marginal likelihood estimate (assumes that bridge estimate is unbiased)
- `percentage`: the approximate percentage error of the marginal likelihood estimate

Note

The following signs can be used to encode restricted hypotheses: `"<"` and `">"` for inequality constraints, `"="` for equality constraints, `","` for free parameters, and `"&"` for independent hypotheses. The restricted hypothesis can either be a string or a character vector. For instance, the hypothesis `c("theta1 < theta2, theta3")` means

- `theta1` is smaller than both `theta2` and `theta3`
- The parameters `theta2` and `theta3` both have `theta1` as lower bound, but are not influenced by each other.

The hypothesis `c("theta1 < theta2 = theta3 & theta4 > theta5")` means that

- Two independent hypotheses are stipulated: `"theta1 < theta2 = theta3"` and `"theta4 > theta5"`
- The restrictions on the parameters `theta1`, `theta2`, and `theta3` do not influence the restrictions on the parameters `theta4` and `theta5`.
- `theta1` is smaller than `theta2` and `theta3`
- `theta2` and `theta3` are assumed to be equal
- `theta4` is larger than `theta5`

References

Damien P, Walker SG (2001). “Sampling truncated normal, beta, and gamma densities.” *Journal of Computational and Graphical Statistics*, **10**, 206–215.

Gronau QF, Sarafoglou A, Matzke D, Ly A, Boehm U, Marsman M, Leslie DS, Forster JJ, Wagenmakers E, Steingroever H (2017). “A tutorial on bridge sampling.” *Journal of Mathematical Psychology*, **81**, 80–97.

Frühwirth-Schnatter S (2004). “Estimating marginal likelihoods for mixture and Markov switching models using bridge sampling techniques.” *The Econometrics Journal*, **7**, 143–167.

Sarafoglou A, Haaf JM, Ly A, Gronau QF, Wagenmakers EJ, Marsman M (2021). “Evaluating Multinomial Order Restrictions with Bridge Sampling.” *Psychological Methods*.

See Also

`generate_restriction_list`

Other functions to evaluate informed hypotheses: `binom_bf_equality()`, `binom_bf_inequality()`, `binom_bf_informed()`, `mult_bf_equality()`, `mult_bf_informed()`

Examples

```
# priors
a <- c(1, 1, 1, 1)

# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

results_prior  <- mult_bf_inequality(Hr=Hr, a=a, factor_levels=factor_levels,
prior=TRUE, seed = 2020, niter = 100)
# corresponds to
cbind(exp(results_prior$logml), 1/factorial(4))

# alternative - if you have samples and a restriction list
inequalities  <- generate_restriction_list(Hr=Hr, a=a,
factor_levels=factor_levels)$inequality_constraints
prior_samples <- mult_tsampling(inequalities, niter = 100,
prior=TRUE, seed = 2020)
results_prior <- mult_bf_inequality(prior_samples, inequalities, niter = 100,
seed=2020)
cbind(exp(results_prior$logml), 1/factorial(4))
```

`mult_bf_informed` *Evaluates Informed Hypotheses on Multinomial Parameters*

Description

Evaluates informed hypotheses on multinomial parameters. These hypotheses can contain (a mixture of) inequality constraints, equality constraints, and free parameters. Informed hypothesis H_r states that category proportions obey the particular constraint. H_r can be tested against the encompassing hypothesis H_e or the null hypothesis H_0. Encompassing hypothesis H_e states that category proportions are free to vary. Null hypothesis H_0 states that category proportions are exactly equal.

Usage

```
mult_bf_informed(
  x,
  Hr,
  a = rep(1, length(x)),
  factor_levels = NULL,
  cred_level = 0.95,
  niter = 5000,
  bf_type = "LogBFer",
  seed = NULL,
  maxiter = 1000,
  nburnin = niter * 0.05
)
```

Arguments

`x`	numeric. Vector with data
`Hr`	string or character. Encodes the user specified informed hypothesis. Use either specified `factor_levels` or indices to refer to parameters. See “Note” section for details on how to formulate informed hypotheses
`a`	numeric. Vector with concentration parameters of Dirichlet distribution. Must be the same length as `x`. Default sets all concentration parameters to 1
`factor_levels`	character. Vector with category names. Must be the same length as `x`
`cred_level`	numeric. Credible interval for the posterior point estimates. Must be a single number between 0 and 1
`niter`	numeric. Vector with number of samples to be drawn from truncated distribution
`bf_type`	character. The Bayes factor type. When the informed hypothesis is compared to the encompassing hypothesis, the Bayes factor type can be `LogBFer`, `BFer`, or `BFre`. When the informed hypothesis is compared to the null hypothesis, the Bayes factor type can be `LogBFr0`, `BF0r`, or `BFr0`. Default is `LogBFer`
`seed`	numeric. Sets the seed for reproducible pseudo-random number generation
`maxiter`	numeric. Maximum number of iterations for the iterative updating scheme used in the bridge sampling routine. Default is 1,000 to avoid infinite loops
`nburnin`	numeric. A single value specifying the number of burn-in samples when drawing from the truncated distribution. Minimum number of burn-in samples is 10. Default is 5% of the number of samples. Burn-in samples are removed automatically after the sampling.

Details

The model assumes that data follow a multinomial distribution and assigns a Dirichlet distribution as prior for the model parameters (i.e., underlying category proportions). That is:

$$x\ Multinomial(N, \theta)$$

$$\theta\ Dirichlet(\alpha)$$

Value

List consisting of the following elements

`$bf_list` gives an overview of the Bayes factor analysis:

- `bf_type`: string. Contains Bayes factor type as specified by the user
- `bf`: data.frame. Contains Bayes factors for all Bayes factor types
- `error_measures`: data.frame. Contains for the overall Bayes factor the approximate relative mean-squared error `re2`, the approximate coefficient of variation `cv`, and the approximate percentage error `percentage`
- `logBFe_equalities`: data.frame. Lists the log Bayes factors for all independent equality constrained hypotheses
- `logBFe_inequalities`: data.frame. Lists the log Bayes factor for all independent inequality constrained hypotheses

`$cred_level` numeric. User specified credible interval

`$restrictions` list that encodes informed hypothesis for each independent restriction:

- `full_model`: list containing the hypothesis, parameter names, data and prior specifications for the full model.
- `equality_constraints`: list containing the hypothesis, parameter names, data and prior specifications for each equality constrained hypothesis.
- `inequality_constraints`: list containing the hypothesis, parameter names, data and prior specifications for each inequality constrained hypothesis. In addition, in `nr_mult_equal` and `nr_mult_free` encodes which and how many parameters are equality constraint or free, in `boundaries` includes the boundaries of each parameter, in `nineq_per_hyp` states the number of inequality constraint parameters per independent inequality constrained hypothesis, and in `direction` states the direction of the inequality constraint.

`$bridge_output` list containing output from bridge sampling function:

- `eval`: list containing the log prior or posterior evaluations (`q11`) and the log proposal evaluations (`q12`) for the prior or posterior samples, as well as the log prior or posterior evaluations (`q21`) and the log proposal evaluations (`q22`) for the samples from the proposal distribution
- `niter`: number of iterations of the iterative updating scheme
- `logml`: estimate of log marginal likelihood
- `hyp`: evaluated inequality constrained hypothesis
- `error_measures`: list containing in `re2` the approximate relative mean-squared error for the marginal likelihood estimate, in `cv` the approximate coefficient of variation for the marginal likelihood estimate (assumes that bridge estimate is unbiased), and in `percentage` the approximate percentage error of the marginal likelihood estimate

`$samples` list containing a list for prior samples and a list of posterior samples from truncated distributions which were used to evaluate inequality constraints. Prior and posterior samples of independent inequality constraints are again saved in separate lists. Samples are stored as matrix of dimension `nsamples x nparams`.

Note

The following signs can be used to encode restricted hypotheses: `"<"` and `">"` for inequality constraints, `"="` for equality constraints, `","` for free parameters, and `"&"` for independent hypotheses. The restricted hypothesis can either be a string or a character vector. For instance, the hypothesis `c("theta1 < theta2, theta3")` means

- `theta1` is smaller than both `theta2` and `theta3`
- The parameters `theta2` and `theta3` both have `theta1` as lower bound, but are not influenced by each other.

The hypothesis `c("theta1 < theta2 = theta3 & theta4 > theta5")` means that

- Two independent hypotheses are stipulated: `"theta1 < theta2 = theta3"` and `"theta4 > theta5"`
- The restrictions on the parameters `theta1`, `theta2`, and `theta3` do not influence the restrictions on the parameters `theta4` and `theta5`.

- `theta1` is smaller than `theta2` and `theta3`
- `theta2` and `theta3` are assumed to be equal
- `theta4` is larger than `theta5`

References

Damien P, Walker SG (2001). "Sampling truncated normal, beta, and gamma densities." *Journal of Computational and Graphical Statistics*, **10**, 206–215.

Gronau QF, Sarafoglou A, Matzke D, Ly A, Boehm U, Marsman M, Leslie DS, Forster JJ, Wagenmakers E, Steingroever H (2017). "A tutorial on bridge sampling." *Journal of Mathematical Psychology*, **81**, 80–97.

Frühwirth-Schnatter S (2004). "Estimating marginal likelihoods for mixture and Markov switching models using bridge sampling techniques." *The Econometrics Journal*, **7**, 143–167.

Sarafoglou A, Haaf JM, Ly A, Gronau QF, Wagenmakers EJ, Marsman M (2021). "Evaluating Multinomial Order Restrictions with Bridge Sampling." *Psychological Methods*.

See Also

Other functions to evaluate informed hypotheses: `binom_bf_equality()`, `binom_bf_inequality()`, `binom_bf_informed()`, `mult_bf_equality()`, `mult_bf_inequality()`

Examples

```
# data
x <- c(3, 4, 10, 11, 7, 30)
# priors
a <- c(1, 1, 1, 1, 1, 1)
# restricted hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4', 'theta5',
'theta6')
Hr            <- c('theta1', '<',  'theta2', '&', 'theta3', '=', 'theta4',
',', 'theta5', '<', 'theta6')
output_total  <- mult_bf_informed(x, Hr, a, factor_levels, seed=2020, niter=100)
```

`mult_tsampling` *Samples From Truncated Dirichlet Density*

Description

Based on specified inequality constraints, samples from truncated prior or posterior Dirichlet density.

Usage

```
mult_tsampling(
  inequalities,
  index = 1,
  niter = 10000,
  prior = FALSE,
  nburnin = niter * 0.05,
  seed = NULL
)
```

Arguments

`inequalities`	list that contains inequality constraints for each independent inequality constrained hypotheses. The list is created in the `generate_restriction_list` function
`index`	numeric. If multiple independent inequality constraints are specified, this index determines for which inequality constraint samples should be drawn. Must be a single value. Default is 1
`niter`	numeric. A single value specifying the number of samples. Default is set to $10,000$
`prior`	logical. If `TRUE` ignores the data that are encoded in `inequalities` and thus samples from the prior distribution. Default is `FALSE`.
`nburnin`	numeric. A single value specifying the number of burn-in samples when drawing from the truncated distribution. Minimum number of burn-in samples is 10. Default is 5% of the number of samples. Burn-in samples are removed automatically after the sampling.
`seed`	numeric. Sets the seed for reproducible pseudo-random number generation

Details

The model assumes that data follow a multinomial distribution and assigns a Dirichlet distribution as prior for the model parameters (i.e., underlying category proportions). That is:

$$x\ Multinomial(N, \theta)$$

$$\theta\ Dirichlet(\alpha)$$

Value

matrix of dimension `niter * nsamples` containing prior or posterior samples from truncated Dirichlet distribution.

Note

When equality constraints are specified in the restricted hypothesis, this function samples from the conditional Dirichlet distribution given that the equality constraints hold.

Only inequality constrained parameters are sampled. Free parameters or parameters that are exclusively equality constrained will be ignored.

References

Damien P, Walker SG (2001). “Sampling truncated normal, beta, and gamma densities.” *Journal of Computational and Graphical Statistics*, **10**, 206–215.

Sarafoglou A, Haaf JM, Ly A, Gronau QF, Wagenmakers EJ, Marsman M (2021). “Evaluating Multinomial Order Restrictions with Bridge Sampling.” *Psychological Methods*.

See Also

`generate_restriction_list`

Examples

```
x <- c(200, 130, 40, 10)
a <- c(1, 1, 1, 1)
factor_levels <- c('mult1', 'mult2', 'mult3', 'mult4')
Hr <- c('mult1 > mult2 > mult3 > mult4')

# generate restriction list
inequalities <- generate_restriction_list(x=x, Hr=Hr, a=a,
factor_levels=factor_levels)$inequality_constraints

# sample from prior distribution
prior_samples <- mult_tsampling(inequalities, niter = 500, prior=TRUE)
# sample from posterior distribution
post_samples <- mult_tsampling(inequalities, niter = 500)
```

`peas` *Mendelian Laws of Inheritance*

Description

This data set, "peas", provides the categorization of crossbreeds between a plant variety that produced round yellow peas with a plant variety that produced wrinkled green peas. This data set contains the categorization of 556 plants that were categorized either as (1) round and yellow, (2) wrinkled and yellow, (3) round and green, or (4) wrinkled and green.

Usage

```
data(peas)
```

Format

A `data.frame` with 4 rows and 2 variables:

`peas`	Crossbreeds that are categorized as ’roundYellow’, ’wrinkledYellow’, ’roundGreen’, or ’wrinkledGreen’.
`counts`	The number of plants assigned to a one of the crossbreed categories.

References

Mulder J, Wagenmakers E, Marsman M (in press). "A Generalization of the Savage-Dickey Density Ratio for Testing Equality and Order Constrained Hypotheses." *The American Statistician*.

Robertson T (1978). "Testing for and against an order restriction on multinomial parameters." *Journal of the American Statistical Association*, **73**, 197–202.

Sarafoglou A, Haaf JM, Ly A, Gronau QF, Wagenmakers EJ, Marsman M (2021). "Evaluating Multinomial Order Restrictions with Bridge Sampling." *Psychological Methods*.

Examples

```
data("peas")
# Prior specification
# We assign a uniform Dirichlet distribution, that is, we set all
# concentration parameters to 1
a <- c(1, 1, 1, 1)

x <- peas$counts
factor_levels <- levels(peas$peas)
# Test the following mixed Hypothesis:
# Hr: roundYellow > wrinkledYellow = roundGreen > wrinkledGreen
#
# Be careful: Factor levels are usually ordered alphabetically!
# When specifying hypotheses using indexes, make sure they refer to the
# correct factor levels.
Hr <- c('1 > 2 = 3 > 4')
# To avoid mistakes, write out factor levels explicitly:
Hr <- c('roundYellow > wrinkledYellow = roundGreen > wrinkledGreen')

out <- mult_bf_informed(x=x, Hr=Hr, a=a, factor_levels=factor_levels, niter=100,
bf_type = 'BFre')
summary(out)
```

`plot.summary.bmult` *Plot estimates*

Description

Plots the posterior estimates from the unconstrained multi- or binomial model.

Usage

```
## S3 method for class 'summary.bmult'
plot(
  x,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  xlim = NULL,
```

```
  ylim = NULL,
  panel.first = NULL,
  ...
)
```

Arguments

`x`	A `summary.bmult`-object returned by `summary()`.
`main`	`character`. A string used as title. Defaults to the informed hypothesis and the Bayes factor.
`xlab`	`character`. A string used as title for the x-axis.
`ylab`	`character`. A string used as title for the y-axis.
`xlim`, `ylim`	numeric vectors of length 2, giving the x and y coordinates ranges.
`panel.first`	`expression`. Evaluated after the plot axes are set up but before any plotting takes place. This can be useful for drawing background grids or null distributions.
`...`	additional arguments passed to `plot.summary.bmult`-method.

Value

Invisibly returns a `data.frame` with the plotted estimates.

Examples

```
# data
x <- c(3, 4, 10, 11, 7, 30)
# priors
a <- c(1, 1, 1, 1, 1, 1)
# restricted hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4', 'theta5',
                   'theta6')
Hr <- c('theta1', '<',  'theta2', '&', 'theta3', '=',
'theta4', ',', 'theta5', '<', 'theta6')
output_total  <- mult_bf_informed(x, Hr, a, factor_levels, seed=2020,
niter=100, bf_type = "BFer")
plot(summary(output_total))

# data for a big Bayes factor
x <- c(3, 4, 10, 11, 7, 30) * 1000
output_total  <- mult_bf_informed(x, Hr, a, factor_levels, seed=2020,
niter=100, bf_type = "BFre")
plot(summary(output_total))
```

`print.bmult` *print method for class* `bmult`

Description

Prints model specification

Usage

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

Arguments

`x`	object of class `bmult` as returned from `mult_bf_informed` or `binom_bf_informed`
`...`	additional arguments, currently ignored

Value

The print methods print the model specifications and descriptives and return nothing

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Binomial Case
out_binom  <- binom_bf_informed(x=x, n=n, Hr=Hr, a=a, b=b, niter=1e3,factor_levels, seed=2020)
out_binom
## Multinomial Case
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, niter=1e3,factor_levels, seed=2020)
out_mult
```

`print.bmult_bridge` *Print method for class* `bmult_bridge`

Description

Prints model specification

Usage

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

Arguments

`x`	object of class `bmult_bridge` as returned from `mult_bf_inequality` or `binom_bf_inequality`
`...`	additional arguments, currently ignored

Value

The print methods print the results from the bridge sampling algorithm and return nothing

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
out_mult  <- mult_bf_inequality(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=1e3, seed=2020)
out_mult
```

`print.summary.bmult` *print method for class* `summary.bmult`

Description

Prints the summary from Bayes factor analysis

Usage

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

Arguments

`x`	object of class `bmult` or `summary.bmult` as returned from `summary.bmult`
`...`	additional arguments, currently ignored

Value

The print methods print the summary from the Bayes factor analysis and returns nothing

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Binomial Case
out_binom  <- binom_bf_informed(x=x, n=n, Hr=Hr, a=a, b=b, niter=1e3,factor_levels, seed=2020)
summary(out_binom)
## Multinomial Case
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, niter=1e3,factor_levels, seed=2020)
summary(out_mult)
```

`print.summary.bmult_bridge`

print method for class `summary.bmult_bridge`

Description

Prints the summary of bridge sampling output

Usage

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

Arguments

`x`	object of class `summary.bmult_bridge` as returned from `summary.bmult_bridge`
`...`	additional arguments, currently ignored

Value

The print methods print the summary of the bridge sampling output

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
out_mult  <- mult_bf_inequality(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=1e3, seed=2020)
summary(out_mult)
```

`restriction_list` *S3 method for class* `restriction_list.bmult`

Description

Extracts restriction list from an object of class `bmult`

Usage

```
restriction_list(x, ...)
```

Arguments

`x`	object of class `bmult` as returned from `mult_bf_informed` or `binom_bf_informed`
`...`	additional arguments passed to other methods

Value

Extracts restriction list and associated hypothesis from an object of class `bmult`

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
```

```
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=1e3, seed=2020)
restriction_list <- restriction_list(out_mult)
```

`restriction_list.bmult` *Extracts restriction list from an object of class* `bmult`

Description

Extracts restriction list from an object of class `bmult`

Usage

```
## S3 method for class 'bmult'
restriction_list(x, restrictions = "inequalities", ...)
```

Arguments

`x`	object of class `bmult` as returned from `mult_bf_informed` or `binom_bf_informed`
`restrictions`	specifies whether to extract restriction list for `equalities` or `inequalities`. Default is `inequalities`.
`...`	additional arguments, currently ignored

Value

Extracts restriction list and associated hypothesis from an object of class `bmult`

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=1e3, seed=2020)
restriction_list <- restriction_list(out_mult)
```

`samples` *S3 method for class 'samples.bmult'*

Description

Extracts prior and posterior samples (if applicable) from an object of class `bmult`

Usage

```
samples(x)
```

Arguments

`x`	object of class `bmult` as returned from `mult_bf_informed` or `binom_bf_informed`

Value

Returns `list` with prior and posterior samples (if applicable) from an object of class `bmult`

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=1e3, seed=2020)
sample_list <- samples(out_mult)
```

`samples.bmult` *Extracts prior and posterior samples (if applicable) from an object of class* `bmult`

Description

Extracts prior and posterior samples (if applicable) from an object of class `bmult`

Usage

```
## S3 method for class 'bmult'
samples(x)
```

Arguments

`x`	object of class `bmult` as returned from `mult_bf_informed` or `binom_bf_informed`

Value

Returns `list` with prior and posterior samples (if applicable) from an object of class `bmult`

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=1e3, seed=2020)
sample_list <- samples(out_mult)
```

`summary.bmult` *summary method for class* `bmult`

Description

Summarizes results from Bayes factor analysis

Usage

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

Arguments

`object`	object of class `bmult` as returned from `mult_bf_informed` or `binom_bf_informed`
`...`	additional arguments, currently ignored

Value

Invisibly returns a `list` which contains the Bayes factor and associated hypotheses for the full model, but also the separate for the independent equality and inequality constraints.

The summary method returns a `list` with the following elements:

`$hyp` Vector containing the informed hypothesis as specified by the user

`$bf` Contains Bayes factor

`$logmlHe` Contains log marginal likelihood of the encompassing model

`$logmlH0` Contains log marginal likelihood of the null model

`$logmlHr` Contains log marginal likelihood of the informed model

`$re2` Contains relative mean-square error for the Bayes factor

`$bf_type` Contains Bayes factor type as specified by the user

`$cred_level` Credible interval for the posterior point estimates.

`$prior` List containing the prior parameters.

`$data` List containing the data.

`$nr_equal` Number of independent equality-constrained hypotheses.

`$nr_inequal` Number of independent inequality-constrained hypotheses.

`$estimates` Parameter estimates for the encompassing model

- `factor_level`: Vector with category names
- `alpha`: Vector with posterior concentration parameters of Dirichlet distribution (for multinomial models) or alpha parameters for independent beta distributions (for binomial models)
- `beta`: Vector with beta parameters for independent beta distributions (for binomial models)
- `lower`: Lower value of credible intervals of marginal beta distributions
- `median`: Posterior median of marginal beta distributions
- `upper`: Upper value of credible intervals of marginal beta distributions

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Binomial Case
out_binom  <- binom_bf_informed(x=x, n=n, Hr=Hr, a=a, b=b, niter=1e3,factor_levels, seed=2020)
summary(out_binom)
## Multinomial Case
out_mult  <- mult_bf_informed(x=x, Hr=Hr, a=a, niter=1e3,factor_levels, seed=2020)
summary(out_mult)
```

`summary.bmult_bridge` *summary method for class* `bmult_bridge`

Description

Summarizes bridge sampling results and associated error measures

Usage

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

Arguments

`object`	object of class `bmult_bridge` as returned from `mult_bf_inequality` or `binom_bf_inequality`
`...`	additional arguments, currently ignored

Value

Invisibly returns a `list` which contains the log marginal likelihood for inequality constrained category proportions and associated error terms.

Examples

```
# data
x <- c(3, 4, 10, 11)
n <- c(15, 12, 12, 12)
# priors
a <- c(1, 1, 1, 1)
b <- c(1, 1, 1, 1)
# informed hypothesis
factor_levels <- c('theta1', 'theta2', 'theta3', 'theta4')
Hr            <- c('theta1', '<',  'theta2', '<', 'theta3', '<', 'theta4')

## Multinomial Case
out_mult  <- mult_bf_inequality(x=x, Hr=Hr, a=a, factor_levels=factor_levels,
niter=1e3, seed=2020)
summary(out_mult)
```

`tbinom_backtrans` *Backtransforms Samples From Real Line To Beta Parameters*

Description

Transforms samples from the real line to samples from a truncated beta density using a stick-breaking algorithm. This algorithm is suitable for mixtures of equality constrained parameters, inequality constrained parameters, and free parameters

Usage

```
tbinom_backtrans(xi_mat, boundaries, binom_equal, hyp_direction)
```

Arguments

`xi_mat`	matrix with samples from truncated beta density. These samples should be transformed, so they range over the entire real line
`boundaries`	list containing indices for upper and lower truncation boundaries
`binom_equal`	multiplicative elements for each lower and upper bound of each inequality constrained parameter.
`hyp_direction`	specifies whether the imposed inequality constrained imposes an increasing (i.e., 'smaller') or decreasing (i.e., 'larger') trend

Value

list consisting of the following elements: (1) theta_mat: matrix with transformed samples (2) lower_mat: matrix containing the lower bound for each parameter (3) upper_mat: matrix containing the upper bound for each parameter

---

`tbinom_trans` *Transforms Truncated Beta Samples To Real Line*

Description

Transforms samples from a truncated beta density to the real line using a stick-breaking algorithm. This algorithm is suitable for mixtures of equality constrained parameters, inequality constrained parameters, and free parameters

Usage

```
tbinom_trans(theta_mat, boundaries, binom_equal, hyp_direction)
```

Arguments

`theta_mat`	matrix with samples from truncated beta density
`boundaries`	list containing indices for upper and lower truncation boundaries
`binom_equal`	multiplicative elements for each lower and upper bound of each inequality constrained parameter.
`hyp_direction`	specifies whether the imposed inequality constrained imposes an increasing (i.e., 'smaller') or decreasing (i.e., 'larger') trend

Value

matrix with transformed samples

`tdir_backtrans` *Backtransforms Samples From Real Line To Dirichlet Parameters*

Description

Transforms samples from the real line to samples from a truncated Dirichlet density using a stick-breaking algorithm. This algorithm is suitable for mixtures of equality constrained parameters, inequality constrained parameters, and free parameters

Usage

```
tdir_backtrans(
  xi_mat,
  boundaries,
  mult_equal,
  nr_mult_equal,
  nr_mult_free,
  hyp_direction
)
```

Arguments

`xi_mat`	matrix with samples from truncated Dirichlet density. These samples should be transformed, so they range over the entire real line
`boundaries`	list containing indices for upper and lower truncation boundaries
`mult_equal`	multiplicative elements for each lower and upper bound of each inequality constrained parameter.
`nr_mult_equal`	vector of multiplicative elements of collapsed parameters
`nr_mult_free`	vector of multiplicative elements of free parameters
`hyp_direction`	specifies whether the imposed inequality constrained imposes an increasing (i.e., 'smaller') or decreasing (i.e., 'larger') trend

Value

list consisting of the following elements: (1) theta_mat: matrix with transformed samples (2) lower_mat: matrix containing the lower bound for each parameter (3) upper_mat: matrix containing the upper bound for each parameter

---

`tdir_trans` *Transforms Truncated Dirichlet Samples To Real Line*

Description

Transforms samples from a truncated Dirichlet density to the real line using a stick-breaking algorithm. This algorithm is suitable for mixtures of equality constrained parameters, inequality constrained parameters, and free parameters

Usage

```
tdir_trans(
  theta_mat,
  boundaries,
  mult_equal,
  nr_mult_equal,
  nr_mult_free,
  hyp_direction
)
```

Arguments

`theta_mat`	matrix with samples from truncated Dirichlet density
`boundaries`	list containing indices for upper and lower truncation boundaries
`mult_equal`	multiplicative elements for each lower and upper bound of each inequality constrained parameter.
`nr_mult_equal`	vector of multiplicative elements of collapsed parameters
`nr_mult_free`	vector of multiplicative elements of free parameters
`hyp_direction`	specifies whether the imposed inequality constrained imposes an increasing (i.e., ’smaller’) or decreasing (i.e., ’larger’) trend

Value

matrix with transformed samples

Index