

Package: `appraise` (via `r-universe`)

May 10, 2026

Title Bias-Aware Evidence Synthesis in Systematic Reviews

Version 0.1.1

Description Implements a bias-aware framework for evidence synthesis in systematic reviews and health technology assessments, as described in Kabali (2025) <[doi:10.1111/jep.70272](https://doi.org/10.1111/jep.70272)>. The package models study-level effect estimates by explicitly accounting for multiple sources of bias through prior distributions and propagates uncertainty using posterior simulation. Evidence across studies is combined using posterior mixture distributions rather than a single pooled likelihood, enabling probabilistic inference on clinically or policy-relevant thresholds. The methods are designed to support transparent decision-making when study relevance and bias vary across the evidence base.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats

Suggests sn, VGAM, cmdstanr, rmarkdown, knitr

Additional_repositories <https://stan-dev.r-universe.dev/>

VignetteBuilder knitr

NeedsCompilation no

Author Conrad Kabali [aut, cre]

Maintainer Conrad Kabali <conrad.kabali@utoronto.ca>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-02-09 21:00:43 UTC

RemoteUrl <https://github.com/cran/appraise>

RemoteRef HEAD

RemoteSha a5d99254da04bb7d56bb9014e8d5af61aabdbc55

Contents

build_bias_specification	2
has_duplicates	4
posterior_mixture	4
posterior_probability	5
posterior_summary	6
run_appraise_model	7
simulate_bias_priors	9
validate_bias_selection	10
validate_positive	11

Index	12
--------------	-----------

build_bias_specification
Build bias specification for Stan

Description

Build bias specification for Stan

Usage

```
build_bias_specification(
  num_biases,
  b_types = character(),
  s_types = character(),
  d_types = character(),
  e_types = character(),
  en_types = character(),
  ab_params = list(),
  skn_params = list(),
  de_params = list(),
  ex_params = list(),
  exneg_params = list()
)
```

Arguments

num_biases	Integer. Total number of biases.
b_types	Character vector of biases with beta priors.
s_types	Character vector of biases with skew-normal priors.
d_types	Character vector of biases with Laplace priors.
e_types	Character vector of biases with exponential priors.
en_types	Character vector of biases with negative exponential priors.

ab_params	Named list of beta prior parameters.
skn_params	Named list of skew-normal prior parameters.
de_params	Named list of Laplace prior parameters.
ex_params	Named list of exponential prior parameters.
exneg_params	Named list of negative exponential prior parameters.

Value

A list defining bias structure and prior parameters.

References

Kabali C (2025). AppRaise: Software for quantifying evidence uncertainty in systematic reviews using a posterior mixture model. *Journal of Evaluation in Clinical Practice*, 31, 1-12. <https://doi.org/10.1111/jep.70272>.

See Also

- [simulate_bias_priors](#) for sampling bias prior distributions
- [run_appraise_model](#) for posterior inference
- `vignette("appraise-introduction")` for a full workflow

Examples

```
## Example 1: Single bias with a Beta prior

bias_spec <- build_bias_specification(
  num_biases = 1,
  b_types = "Confounding",
  ab_params = list(
    Confounding = c(2, 5)
  )
)

bias_spec

## Example 2: Multiple biases with different prior families

bias_spec <- build_bias_specification(
  num_biases = 2,
  b_types = "Confounding",
  s_types = "Selection Bias",
  ab_params = list(
    Confounding = c(2, 5)
  ),
  skn_params = list(
    `Selection Bias` = c(0, 0.2, 5)
  )
)

bias_spec
```

```
## Example 3: Exponential bias prior

bias_spec <- build_bias_specification(
  num_biases = 1,
  e_types = "Measurement Errors",
  ex_params = list(
    `Measurement Errors` = 1.5
  )
)

bias_spec
```

has_duplicates	<i>Check for duplicate values</i>
----------------	-----------------------------------

Description

Check for duplicate values

Usage

```
has_duplicates(x)
```

Arguments

x	Character vector
---	------------------

Value

Logical. TRUE if duplicates exist.

posterior_mixture	<i>Posterior mixture across studies</i>
-------------------	---

Description

Combines posterior draws across studies using a weighted mixture

Usage

```
posterior_mixture(theta_list, weights)
```

Arguments

theta_list	List of numeric vectors of posterior draws
weights	Numeric vector of study weights

Value

A list containing mixture draws and summaries

References

Kabali C (2025). AppRaise: Software for quantifying evidence uncertainty in systematic reviews using a posterior mixture model. *Journal of Evaluation in Clinical Practice*, 31, 1-12. <https://doi.org/10.1111/jep.70272>.

See Also

- [posterior_probability](#)
- [posterior_summary](#)
- `vignette("appraise-introduction")`

Examples

```
# Simulate posterior draws from two studies
theta1 <- rnorm(1000, mean = -0.6, sd = 0.1)
theta2 <- rnorm(1000, mean = -0.4, sd = 0.15)

# Combine using relevance weights
mix <- posterior_mixture(
  theta_list = list(theta1, theta2),
  weights = c(0.6, 0.4)
)

# Mixture draws
head(mix$draws)

# Posterior summary (mean and 95% credible interval)
mix$summary
```

posterior_probability *Posterior probability of significance*

Description

Computes the posterior probability that the target parameter exceeds the threshold for significance.

Usage

```
posterior_probability(mid_samples)
```

Arguments

mid_samples Numeric vector of indicator draws (0/1)

Value

Numeric probability

References

Kabali C (2025). AppRaise: Software for quantifying evidence uncertainty in systematic reviews using a posterior mixture model. *Journal of Evaluation in Clinical Practice*, 31, 1-12. <https://doi.org/10.1111/jep.70272>.

See Also

- [posterior_mixture](#)
- `vignette("appraise-introduction")`

Examples

```
# Simulated posterior draws
set.seed(123)
mid_samples <- rnorm(2000, mean = -0.3, sd = 0.1)

# Posterior probability of benefit
posterior_probability(mid_samples)
```

posterior_summary *Posterior summary statistics*

Description

Computes posterior mean and credible interval

Usage

```
posterior_summary(theta_samples, probs = c(0.025, 0.975))
```

Arguments

theta_samples Numeric vector of posterior draws
probs Credible interval probabilities

Value

Named numeric vector

References

Kabali C (2025). AppRaise: Software for quantifying evidence uncertainty in systematic reviews using a posterior mixture model. *Journal of Evaluation in Clinical Practice*, 31, 1-12. <https://doi.org/10.1111/jep.70272>

See Also

- [posterior_mixture](#)
- `vignette("appraise-introduction")`

Examples

```
# Simulated posterior draws for a treatment effect
set.seed(123)
theta_samples <- rnorm(2000, mean = -0.4, sd = 0.15)

# Posterior mean and 95% credible interval
posterior_summary(theta_samples)

# Custom credible interval (e.g., 90%)
posterior_summary(theta_samples, probs = c(0.05, 0.95))
```

run_appraise_model *Run the appraise Stan model*

Description

Executes the posterior mixture model described in Kabali (2025)

Usage

```
run_appraise_model(
  bias_spec,
  yhat,
  stdev,
  threshold_value,
  iter_sampling = 5000,
  iter_warmup = 1000,
  chains = 4,
  seed = 12345
)
```

Arguments

bias_spec	Output from <code>build_bias_specification()</code>
yhat	Reported point estimate
stdev	Reported standard error
threshold_value	Threshold for significance
iter_sampling	Number of sampling iterations
iter_warmup	Number of warmup iterations
chains	Number of MCMC chains
seed	Random seed

Value

A list containing the CmdStan fit object and posterior draws

References

Kabali C (2025). AppRaise: Software for quantifying evidence uncertainty in systematic reviews using a posterior mixture model. *Journal of Evaluation in Clinical Practice*, 31, 1-12. <https://doi.org/10.1111/jep.70272>.

See Also

- [build_bias_specification](#)
- [posterior_probability](#)
- [posterior_summary](#)
- `vignette("appraise-introduction")`

Examples

```
# Define a simple bias specification with one bias
bias_spec <- build_bias_specification(
  num_biases = 1,
  b_types = "Confounding",
  ab_params = list(
    Confounding = c(2, 5)
  )
)

bias_spec

# Run the AppRaise model for a single study. Requires cmdstanr and a CmdStan
# installation.
fit <- run_appraise_model(
  bias_spec = bias_spec,
  yhat = -0.6,
  stdev = 0.12,
  threshold_value = -0.4,
  iter_sampling = 500,
  iter_warmup = 250,
  chains = 2,
  seed = 123
)

# Posterior draws of the causal effect
head(fit$theta)

# Posterior probability of exceeding the threshold
posterior_probability(fit$mid)

# Posterior summary
posterior_summary(fit$theta)
```

simulate_bias_priors *Simulate bias priors (xi)*

Description

Generates Monte Carlo samples from the prior distributions specified for each bias type.

Usage

```
simulate_bias_priors(bias_spec, n_draws = 5000)
```

Arguments

bias_spec	Output from build_bias_specification()
n_draws	Number of Monte Carlo draws

Value

A numeric matrix of dimension n_draws x NN

References

Kabali C (2025). AppRaise: Software for quantifying evidence uncertainty in systematic reviews using a posterior mixture model. *Journal of Evaluation in Clinical Practice*, 31, 1-12. <https://doi.org/10.1111/jep.70272>.

See Also

- [build_bias_specification](#)
- [run_appraise_model](#)
- `vignette("appraise-introduction")`

Examples

```
## Simulate prior draws for two biases

bias_spec <- build_bias_specification(
  num_biases = 2,
  b_types = "Confounding",
  e_types = "Measurement Errors",
  ab_params = list(
    Confounding = c(2, 5)
  ),
  ex_params = list(
    `Measurement Errors` = 1.2
  )
)

xi <- simulate_bias_priors(bias_spec, n_draws = 1000)
```

```
## Dimensions correspond to (draws × biases)
dim(xi)

## Inspect prior distribution for first bias
hist(xi[, 1], main = "Prior for Confounding Bias", xlab = "Bias magnitude")
```

```
validate_bias_selection
      Validate bias selections
```

Description

Ensures that the number of selected biases matches the declared total and that no bias is assigned more than one distribution.

Usage

```
validate_bias_selection(
  b_types,
  s_types,
  d_types,
  e_types,
  en_types,
  num_biases
)
```

```
validate_bias_selection(
  b_types,
  s_types,
  d_types,
  e_types,
  en_types,
  num_biases
)
```

Arguments

b_types, s_types, d_types, e_types, en_types
Character vectors of bias names

num_biases Integer. Total number of biases declared.

Value

Invisibly TRUE, otherwise errors

Invisibly TRUE; otherwise throws an error

validate_positive	<i>Validate positivity constraints</i>
-------------------	--

Description

Checks that numeric inputs are strictly positive, allowing for specified index exceptions.

Usage

```
validate_positive(values, exceptions = NULL)
```

```
validate_positive(values, exceptions = NULL)
```

Arguments

values Numeric vector

exceptions Optional integer vector of indices allowed to be non-positive

Value

Invisibly TRUE; otherwise throws an error

Index

`build_bias_specification`, [2](#), [8](#), [9](#)

`has_duplicates`, [4](#)

`posterior_mixture`, [4](#), [6](#), [7](#)

`posterior_probability`, [5](#), [5](#), [8](#)

`posterior_summary`, [5](#), [6](#), [8](#)

`run_appraise_model`, [3](#), [7](#), [9](#)

`simulate_bias_priors`, [3](#), [9](#)

`validate_bias_selection`, [10](#)

`validate_positive`, [11](#)