Package: qbaconfound (via r-universe)

May 26, 2026

Title Monte Carlo Quantitative Bias Analysis for Unmeasured Confounding

Version 0.0.0.9000

Description A flexible Monte Carlo quantitative bias analysis (QBA) for unmeasured confounding in observational studies, as described in Hughes et al. The substantive analysis may be a generalised linear model or a Cox proportional hazards model with a binary, continuous, or categorical exposure and measured confounders. The method allows for one or more binary or continuous unmeasured confounders that may be correlated with the measured confounders. Informative priors for a small number of bias parameters encode external information about the unmeasured confounders.

License MIT + file LICENSE

URL https://github.com/remlapmot/qbaconfound, https://mrcieu.r-universe.dev/qbaconfound

BugReports https://github.com/remlapmot/qbaconfound/issues

Imports stats, survival

Suggests testthat (>= 3.0.0)

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Encoding UTF-8

Repository https://mrcieu.r-universe.dev

Date/Publication 2026-05-26 09:58:40 UTC

RemoteUrl https://github.com/remlapmot/qbaconfound

RemoteRef HEAD

RemoteSha 3ae80e85354f1caf826a089bbd78474b1fcdf5af

Contents

qbaconfound 2
sim_confounding 4
summary.qbaconfound 5
u_binary 5
u_continuous 6

Index **8**

`qbaconfound` *Monte Carlo quantitative bias analysis for unmeasured confounding*

Description

Conducts the flexible Monte Carlo quantitative bias analysis (QBA) for unmeasured confounding of Hughes et al. Given a naive analysis model (which omits one or more unmeasured confounders) and informative priors for a small number of bias parameters, the function returns a bias-adjusted estimate of the exposure effect together with an interval that accounts for both the unmeasured confounding and sampling variability.

Usage

```
qbaconfound(
  formula,
  data,
  exposure = NULL,
  confounders,
  family = stats::gaussian(),
  reps = 1000L,
  sampling_error = TRUE,
  seed = NULL
)
```

Arguments

`formula`	The naive analysis model, e.g. `y ~ x + c1 + c2` for a GLM or `Surv(time, status) ~ x + c1` for a Cox model. The unmeasured confounders are *not* included in `formula`.
`data`	A data frame containing the outcome, exposure, and measured confounders.
`exposure`	Character vector naming the exposure term(s) in `formula` whose effect is of interest. Defaults to the first term on the right-hand side.
`confounders`	A single `u_continuous()`/`u_binary()` object, or a list of them, describing the unmeasured confounder(s).
`family`	The outcome model family: a `stats::glm()` family object, a family name, or `"cox"` for a Cox proportional hazards model. Ignored (and inferred as Cox) when the response is a `survival::Surv()` call.

reps	Number of Monte Carlo replications.
sampling_error	Logical; if TRUE (the default) Monte Carlo sampling error is incorporated at step 4 above. Set to FALSE to obtain the distribution of bias-adjusted point estimates without sampling error.
seed	Optional integer seed for reproducibility.

Details

The substantive analysis may be a generalised linear model (any stats::glm() family) or a Cox proportional hazards model. Survival outcomes are detected automatically when the left-hand side of formula is a survival::Surv() call, or family can be set to "cox".

Each Monte Carlo replication (see Hughes et al., section 2.4):

1. draws a value for every bias parameter from its prior (the priors are specified through u_continuous() / u_binary());
2. simulates a proxy for each unmeasured confounder as a function of the exposure only (a continuous proxy from a normal model, a binary proxy from a Bernoulli model whose intercept reproduces the drawn prevalence);
3. refits the outcome model including the simulated proxies, with their coefficients fixed to the drawn values (implemented as a model offset), and reads off the exposure coefficient and its standard error;
4. adds Monte Carlo sampling error by drawing the bias-adjusted estimate from a normal distribution centred on that coefficient.

The point estimate is the median and the interval the 2.5th and 97.5th percentiles of the resulting distribution of bias-adjusted estimates.

Value

An object of class qbaconfound, a list with elements including estimates (a data frame of naive and bias-adjusted estimates per exposure term), draws (the matrix of bias-adjusted estimates across replications), and n_failed (the number of replications whose model fit failed).

References

Hughes RA, Kawabata E, Palmer TM, et al. A flexible Monte Carlo quantitative bias analysis for unmeasured confounding. *Statistical Methods in Medical Research* (under review).

See Also

u_continuous(), u_binary(), sim_confounding()

Examples

```
df <- sim_confounding(n = 500, beta_x = 0, seed = 1)

# Naive model y ~ x + c1 omits the confounder u; adjust for one continuous U.
fit <- qbaconfound(
  y ~ x + c1, data = df, exposure = "x",
```

```
  confounders = u_continuous(coef_out = c(0.8, 0.1),
                             coef_exp = c(0.6, 0.05),
                             resid_sd = c(0.9, 1.1)),
  reps = 200, seed = 1
)
fit
```

`sim_confounding` *Simulate data with an unmeasured confounder*

Description

A small helper that simulates a dataset with one measured confounder `c1` and one unmeasured confounder `u` that jointly confound the exposure-outcome relationship. The naive model `y ~ x + c1` (which omits `u`) is therefore biased for the exposure effect, making the data useful for examples and tests of `qbaconfound()`.

Usage

```
sim_confounding(
  n = 1000L,
  beta_x = 0,
  family = c("gaussian", "binomial"),
  seed = NULL
)
```

Arguments

`n`	Number of observations.
`beta_x`	True exposure effect (on the linear-predictor scale).
`family`	Outcome type: `"gaussian"` (continuous outcome) or `"binomial"` (binary outcome).
`seed`	Optional integer seed for reproducibility.

Value

A data frame with columns `y`, `x`, `c1`, and `u` (the unmeasured confounder, included so it can be removed to mimic the unmeasured case). The true exposure effect is stored in `attr(, "beta_x")`.

Examples

```
df <- sim_confounding(n = 1000, beta_x = 0.5, seed = 42)
# Naive (biased) versus full (unbiased) model:
coef(lm(y ~ x + c1, df))["x"]
coef(lm(y ~ x + c1 + u, df))["x"]
```

summary.qbaconfound *Summarise a Monte Carlo QBA*

Description

Summarise a Monte Carlo QBA

Usage

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

Arguments

`object`	A `qbaconfound` object returned by `qbaconfound()`.
`...`	Unused.

Value

A data frame of the naive and bias-adjusted estimates for each exposure term, with columns `term`, `naive`, `naive_se`, `estimate`, `conf.low`, and `conf.high`.

u_binary *Specify a binary unmeasured confounder*

Description

Describes the prior distributions of the bias parameters for a single binary unmeasured confounder, for use in `qbaconfound()`.

Usage

```
u_binary(coef_out, coef_exp, prevalence, prev_dist = c("uniform", "beta"))
```

Arguments

`coef_out`	Length-2 numeric `c(mean, sd)` giving the normal prior for the coefficient of the unmeasured confounder in the outcome model (β_U). Set `sd = 0` for a point-mass (fixed value) prior.
`coef_exp`	Normal prior for the coefficient(s) of the exposure in the model for the unmeasured confounder (α_X). For a single exposure term, a length-2 numeric `c(mean, sd)`. For a categorical exposure with several terms, a matrix with one row `c(mean, sd)` per exposure term, or a list of such length-2 vectors.
`prevalence`	Length-2 numeric giving the prior for the marginal prevalence π of the binary unmeasured confounder. If `prev_dist = "uniform"` (the default) this is `c(min, max)` of a uniform prior; if `prev_dist = "beta"` this is `c(a, b)` of a beta prior.
`prev_dist`	Prior family for the prevalence: `"uniform"` or `"beta"`.

Details

As for `u_continuous()` the coefficient of `U` in the outcome model (`coef_out`) and the coefficient(s) of the exposure in the model for `U` (`coef_exp`) are bias parameters. For a binary confounder the third bias parameter is the marginal prevalence of `U` (`prevalence`, i.e. π) rather than a residual standard deviation. A prevalence is usually easier to elicit and more readily reported in the literature than the intercept of a logistic model; the intercept needed to reproduce the drawn prevalence is derived internally.

Value

An object of class `qba_u` describing a binary unmeasured confounder.

See Also

`u_continuous()`, `qbaconfound()`

Examples

```
u_binary(coef_out = c(0.7, 0.1), coef_exp = c(0.4, 0.05),
         prevalence = c(0.15, 0.25))
```

---

u_continuous — *Specify a continuous unmeasured confounder*

Description

Describes the prior distributions of the bias parameters for a single continuous unmeasured confounder, for use in `qbaconfound()`.

Usage

```
u_continuous(coef_out, coef_exp, resid_sd, resid_dist = c("uniform", "gamma"))
```

Arguments

`coef_out`	Length-2 numeric `c(mean, sd)` giving the normal prior for the coefficient of the unmeasured confounder in the outcome model (β_U). Set `sd = 0` for a point-mass (fixed value) prior.
`coef_exp`	Normal prior for the coefficient(s) of the exposure in the model for the unmeasured confounder (α_X). For a single exposure term, a length-2 numeric `c(mean, sd)`. For a categorical exposure with several terms, a matrix with one row `c(mean, sd)` per exposure term, or a list of such length-2 vectors.
`resid_sd`	Length-2 numeric giving the prior for the residual standard deviation η. If `resid_dist = "uniform"` (the default) this is `c(min, max)` of a uniform prior on η. If `resid_dist = "gamma"` this is `c(shape, scale)` of a gamma prior on the residual *precision* $1/\eta^2$.
`resid_dist`	Prior family for the residual standard deviation: either `"uniform"` (on η) or `"gamma"` (on the precision $1/\eta^2$).

Details

The bias model relates the unmeasured confounder `U` to the study data through three bias parameters: the coefficient of `U` in the outcome model (`coef_out`, i.e. β_U), the coefficient(s) of the exposure in the model for `U` (`coef_exp`, i.e. α_X), and the residual standard deviation of `U` given the exposure (`resid_sd`, i.e. η). Values for these parameters cannot be estimated from the data and so are drawn from the prior distributions specified here.

Value

An object of class `qba_u` describing a continuous unmeasured confounder.

See Also

`u_binary()`, `qbaconfound()`

Examples

```
u_continuous(coef_out = c(0.8, 0.1), coef_exp = c(0.3, 0.05),
             resid_sd = c(0.9, 1.1))
```

Index

qbaconfound, 2
qbaconfound(), *4–7*

sim_confounding, 4
sim_confounding(), *3*
stats::glm(), *2, 3*
summary.qbaconfound, 5
survival::Surv(), *2, 3*

u_binary, 5
u_binary(), *2, 3*, *7*
u_continuous, 6
u_continuous(), *2, 3*, *6*