Package 'ameras'

March 29, 2026

Title Analyze Multiple Exposure Realizations in Association Studies

Version 0.1.1

Depends R (>= 3.5.0), stats, nimble

Suggests knitr, rmarkdown, testthat (>= 3.0.0), ggplot2

Description Analyze association studies with multiple realizations of a noisy or uncertain exposure. These can be obtained from e.g. a two-dimensional Monte Carlo dosimetry system (Simon et al 2015 <doi:10.1667/RR13729.1>) to characterize exposure uncertainty. The implemented methods are regression calibration (Carroll et al. 2006 <doi:10.1201/9781420010138>), extended regression calibration (Little et al. 2023 <doi:10.1038/s41598-023-42283-y>), Monte Carlo maximum likelihood (Stayner et al. 2007 <doi:10.1667/RR0677.1>), frequentist model averaging (Kwon et al. 2023 <doi:10.1371/journal.pone.0290498>), and Bayesian model averaging (Kwon et al. 2016 <doi:10.1002/sim.6635>). Supported model families are Gaussian, binomial, multinomial, Poisson, proportional hazards, and conditional logistic.

License MIT + file LICENSE

Imports Rcpp (>= 1.0.10), RcppEigen, coda, numDeriv, MCMCvis, mvtnorm, memoise, methods

LinkingTo Rcpp, RcppEigen

NeedsCompilation yes

VignetteBuilder knitr

Config/testthat/edition 3

Author Sander Roberti [aut, cre] (ORCID: <https://orcid.org/0000-0002-6275-7442>), William Wheeler [aut], Deukwoo Kwon [aut] (ORCID: <https://orcid.org/0000-0001-5376-5320>), Ruth Pfeiffer [ctb] (ORCID: <https://orcid.org/0000-0001-7791-2698>), NCI [cph, fnd]

Maintainer Sander Roberti <sander.roberti@nih.gov>

Repository CRAN

Date/Publication 2026-03-29 16:40:16 UTC

Contents

ameras-package 2
ameras 3
data 8
traceplot 8
transform1 9
transform1.inv 10
transform1.jacobian 10

Index **12**

`ameras-package` *Analyze multiple exposure realizations in association studies*

Description

Analyze association studies with multiple realizations of a noisy or uncertain exposure. These can be obtained from e.g. a two-dimensional Monte Carlo dosimetry system (Simon et al 2015 <doi:10.1667/RR13729.1>) to characterize exposure uncertainty. Methods include regression calibration (Carroll et al. 2006 doi:10.1201/9781420010138), extended regression calibration (Little et al. 2023 doi:10.1038/s4159802342283y), Monte Carlo maximum likelihood (Stayner et al. 2007 doi:10.1667/RR0677.1), frequentist model averaging (Kwon et al. 2023 doi:10.1371/journal.pone.0290498), and Bayesian model averaging (Kwon et al. 2016 doi:10.1002/sim.6635). Supported model families are Gaussian, binomial, multinomial, Poisson, proportional hazards, and conditional logistic.

Details

The main function is `ameras`.

Author(s)

Sander Roberti <sander.roberti@nih.gov>, William Wheeler <WheelerB@imsweb.com>, Ruth Pfeiffer <pfeiffer@mail.nih.gov>, and Deukwoo Kwon <DKwon@uams.edu

References

Roberti, S., Kwon D., Wheeler W., Pfeiffer R. (in preparation). ameras: An R Package to Analyze Multiple Exposure Realizations in Association Studies

`ameras` *Analyze multiple exposure realizations*

Description

Fit regression models accounting for exposure uncertainty using multiple Monte Carlo exposure realizations. Six outcome model families are supported. The first is the Gaussian family for continuous outcomes,

$$Y_i \sim N(\mu_i, \sigma^2),$$

with $\mu_i = \alpha_0 + \boldsymbol{X}_i^T\boldsymbol{\alpha} + \beta_1 D_i + \beta_2 D_i^2 + \boldsymbol{M}_i^T\boldsymbol{\beta}_{m1} D_i + \boldsymbol{M}_i^T\boldsymbol{\beta}_{m2} D_i^2$. Here $\boldsymbol{X}_i$ are covariates, D_i is the exposure with measurement error, and $\boldsymbol{M}_i$ are binary effect modifiers. The quadratic exposure terms and effect modification are optional.

For non-Gaussian families, three relative risk models for the main exposure are supported, the usual exponential $RR_i = \exp(\beta_1 D_i + \beta_2 D_i^2 + \boldsymbol{M}_i^T\boldsymbol{\beta}_{m1} D_i + \boldsymbol{M}_i^T\boldsymbol{\beta}_{m2} D_i^2)$ and the linear excess relative risk (ERR) model $RR_i = 1 + \beta_1 D_i + \beta_2 D_i^2 + \boldsymbol{M}_i^T\boldsymbol{\beta}_{m1} D_i + \boldsymbol{M}_i^T\boldsymbol{\beta}_{m2} D_i^2$, where the quadratic and effect modification terms are optional. Finally, the linear-exponential relative risk model $RR_i = 1 + (\beta_1 + \boldsymbol{M}_i^T\boldsymbol{\beta}_{m1}) D_i \exp\{(\beta_2 + \boldsymbol{M}_i^T\boldsymbol{\beta}_{m2}) D_i\}$ is supported.

The second supported family is logistic regression for binary outcomes, with probabilities

$$p_i/(1-p_i) = RR_i \exp(\alpha_0 + \boldsymbol{X}_i^T\boldsymbol{\alpha}).$$

Third is Poisson regression for counts,

$$Y_i \sim \text{Poisson}(\mu_i),$$

where $\mu_i = RR_i \exp(\alpha_0 + \boldsymbol{X}_i^T\boldsymbol{\alpha}) \times \text{offset}_i$ with optional offset.

Fourth is proportional hazards regression for time-to-event data, with hazard function

$$h(t) = h_0(t) RR_i \exp(\boldsymbol{X}_i^T\boldsymbol{\alpha}),$$

with h_0 the baseline hazard.

Fifth is multinomial logistic regression for a categorical outcome with $Z > 2$ outcome categories, with the last category as the referent category (i.e., $\alpha_{0,Z} = \boldsymbol{\alpha}_Z = \beta_{1,Z} = \beta_{2,Z} = \boldsymbol{\beta}_{m1,Z} = \boldsymbol{\beta}_{m2,Z} = 0$):

$$P(Y_i = z) = RR_i \exp(\alpha_{0,z} + \boldsymbol{X}_i^T\boldsymbol{\alpha}_z) / \{1 + \sum_{s=1}^{Z-1} RR_i \exp(\alpha_{0,s} + \boldsymbol{X}_i^T\boldsymbol{\alpha}_s)\}$$

Sixth is conditional logistic regression for matched case control data, for which

$$P\left(Y_i = 1, Y_k = 0 \forall k \neq i \middle| \sum_{i \in \mathcal{R}} Y_i = 1\right) = RR_i \exp(\boldsymbol{X}_i^T\boldsymbol{\alpha}) / \{\sum_{k \in \mathcal{R}} RR_k \exp(\boldsymbol{X}_k^T\boldsymbol{\alpha})\},$$

where $\mathcal{R}$ is the matched set corresponding to individual i.

Methods include regression calibration (Carroll et al. 2006 doi:10.1201/9781420010138), extended regression calibration (Little et al. 2023 doi:10.1038/s41598023422283y), Monte Carlo maximum likelihood (Stayner et al. 2007 doi:10.1667/RR0677.1), frequentist model averaging (Kwon et al. 2023 doi:10.1371/journal.pone.0290498), and Bayesian model averaging (Kwon et al. 2016 doi:10.1002/sim.6635).

Usage

```
ameras(data, family="gaussian", Y, dosevars, M=NULL, X=NULL, offset=NULL, entry=NULL,
  exit=NULL, setnr=NULL, methods="RC", deg=1, doseRRmod="ERR", transform=NULL,
  transform.jacobian=NULL, inpar=NULL, CI=c("proflik","percentile"),
 params.profCI="dose", maxit.profCI=20, tol.profCI=1e-2, loglim=1e-30, MFMA=100000,
  prophaz.numints.BMA=10, ERRprior.BMA="doubleexponential", nburnin.BMA=5000,
 niter.BMA=20000, nchains.BMA=2, thin.BMA=10, included.replicates.BMA=1:length(dosevars),
  optim.method="Nelder-Mead", control=NULL, ... )
```

Arguments

data	input data frame.
family	outcome model family: "gaussian", "binomial", "poisson", "prophaz", "multinomial" or "clogit" (default "gaussian").
Y	name or column index of the outcome variable for linear, binomial, Poisson, multinomial and conditional logistic models, or event indicator variable for the proportional hazards model.
dosevars	names or column indices of exposure replicate vectors.
M	names or column indices of binary effect modifying variables (optional).
X	names or column indices of other covariates (optional).
offset	name or column index of offset variable for Poisson regression (optional).
entry	name or column index of left truncation time variable for proportional hazards regression (optional).
exit	name or column index of exit time variable, required when family=prophaz.
setnr	name or column index of integer-valued matched set variable, required when family="clogit".
methods	character vector of one or multiple methods to apply. Options: "RC", "ERC", "MCML", "FMA", "BMA" (default "RC").
deg	for doseRRmod="ERR" and doseRRmod="EXP", whether to fit a linear (deg=1) or linear-quadratic (deg=2) dose-response model (default linear).
doseRRmod	the functional form of the dose-response relationship; options are exponential RR ("EXP"), linear ERR ("ERR"), or linear-exponential RR ("LINEXP") (default "ERR").
transform	function for internal parameter transformation (see Details).
transform.jacobian	Jacobian of the transformation function (see Details).
inpar	vector of initial values for log-likelihood optimization (optional).
CI	method for calculation of 95% confidence or credible intervals (see Details). For RC, ERC, and MCML, options are "wald.orig", "wald.transformed", "proflik" (default "proflik"). For FMA and BMA, options are "percentile" and "hpd" (default "percentile"). If methods contains at least one of RC, ERC, and MCML and at least one of FMA and BMA, CI must be length 2 and specify one method for RC, ERC, and MCML, and one for FMA and BMA (see Details).

params.profCI	when CI="proflik", whether to obtain profile-likelihood CIs for all parameters ("all") or only dose-related parameters ("dose", default).
maxit.profCI	maximum iterations for determining profile-likelihood CIs; passed to uniroot (default 20).
tol.profCI	tolerance for determining profile-likelihood CIs; passed to uniroot (default 1e-2).
loglim	parameter used in likelihood computations to avoid taking the log of very small or negative numbers via log(max(x, loglim)) (default 1e-30).
MFMA	number of samples for "FMA" to compute estimates and CIs (default 100,000).
prophaz.numints.BMA	for methods="BMA" with family="prophaz", the number of subintervals with constant baseline hazard (default 10). Cut points are determined based on quantiles of the event time distribution among cases.
ERRprior.BMA	prior for dose-related parameters when doseRRmod="ERR" or "LINEXP" and methods="BMA". Options: "truncated_normal", "truncated_horseshoe", "truncated_doubleexponential", "normal", "horseshoe", "doubleexponential", see Details (default "doubleexponential").
nburnin.BMA	number of MCMC burn-in iterations for BMA (default 1,000).
niter.BMA	number of MCMC iterations per chain for BMA (default 5,000).
nchains.BMA	number of MCMC chains for BMA (default 2).
thin.BMA	thinning rate for BMA (default 10).
included.replicates.BMA	indices of exposure replicates used in BMA (default \ 1:length(dosevars)).
optim.method	method used for optimization by optim. Options are "Nelder-Mead" and "BFGS". When using Nelder-Mead, a second optimization with BFGS is run to ensure an optimal fit.
control	control list passed to optim (default list(reltol=1e-10)).
...	other arguments, passed to functions such as transform.

Details

A transformation can be used to reparametrize parameters internally (i.e., such that the likelihoods are evaluated at transform(parameters), where parameters are unconstrained), and should be specified when fitting linear excess relative risk and linear-exponential models to ensure nonnegative odds/risk/hazard. The included function transform1 applies an exponential transformation to the desired parameters, see ?transform1. When supplying a function to transform, this should be a function of the full parameter vector, returning a full (transformed) parameter vector. In particular, the full parameter vector contains parameters in the following order: $\alpha_0, \boldsymbol{\alpha}, \beta_1, \beta_2, \boldsymbol{\beta}_{m1}, \boldsymbol{\beta}_{m2}, \sigma$, where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}_{m1}$ and $\boldsymbol{\beta}_{m2}$ can be vectors, with lengths matching $\boldsymbol{X}$ and $\boldsymbol{M}$, respectively. σ is only included for the linear model (Gaussian family), and no intercept is included for the proportional hazards and conditional logistic models. For the multinomial model, the full parameter vector is the concatenation of $Z - 1$ parameter vectors in the order as given above, where Z is the number of outcome categories, with the last category chosen as the referent category. See vignette("transformations", package="ameras") for an example of how to specify a custom transformation function.

When no transformation is specified and the linear ERR model is used, `transform1` is used for ERR parameters β_1 and β_2 by default, with lower limits $-1/max(D)$ for β_1 in the linear dose-response and $(0, -1/max(D^2))$ for (β_1, β_2) in the linear-quadratic dose-response, respectively. For the linear-exponential model, a lower limit of 0 is used for β_1, and no transformation is used for β_2. If effect modifiers `M` are specified, no transformation is used for those parameters. When negative RRs are obtained during optimization, an error will be generated and a different transformation or bounds should be used. All output is returned in the original parametrization. The Jacobian of the transformation (`transform.jacobian`) is required when using a transformation. For transform1, the Jacobian is given by transform1.jacobian. No transformations are used in BMA, and FMA is applied on the parameters using the parametrization as given in above with variances obtained using the delta method with the provided Jacobian function.

Multiple options for confidence intervals are provided. For (extended) regression calibration and Monte Carlo maximum likelihood, Wald and profile likelihood intervals can be obtained. When a parameter transformation $\boldsymbol{\theta} = h(\boldsymbol{\eta})$ is used, `CI="wald.transformed"` yields the CI $h(\boldsymbol{\eta} \pm 1.96\boldsymbol{V})$ with $\boldsymbol{V}$ the vector of standard deviations estimated using the inverse Hessian matrix, and `CI="wald.orig"` uses the delta method to obtain the CI $h(\boldsymbol{\eta}) \pm 1.96\boldsymbol{V}_*$ where $\boldsymbol{V}_*$ is the vector of standard deviations estimated using $JH^{-1}J^T$ with J the Jacobian of the transformation and H is the Hessian. When no transformation is used, `CI="wald.orig"` should be used. The third option is `proflik`, which uses the profile likelihood to compute confidence bounds. For FMA and BMA, the options for confidence/credible intervals are `CI="percentile"` which uses 2.5% and 97.5% percentiles, and `CI="hpd"` which computes highest posterior density intervals using `HPDinterval` from the `coda` package, both using the FMA samples or Bayesian posterior samples.

For BMA, a prior distribution for exposure-response parameters can be chosen when using linear or linear-exponential exposure-response model. The options are normal, horshoe, and double exponential priors, and the same priors truncated at 0 to yield positive values. In particular:

- Normal: $\beta_j \sim N(0, 1000)$ for all exposure-response parameters β_j
- Horseshoe (shrinkage prior): $\tau \sim$ Cauchy$(0,1)^+; \lambda_j \sim$ Cauchy$(0,1)^+; \beta_j \sim N(0, \tau^2\lambda_j^2)$. Here τ is shared across all parameters
- Double exponential (shrinkage prior): $\lambda_j \sim$ Cauchy$(0,1)^+; \beta_j \sim$ DoubleExponential$(0, \lambda_j)$

For all other parameters, and when using the exponential exposure-response model or the Gaussian outcome family, the prior is $N(0, 1000)$. For the parameter σ in the Gaussian family, this prior is truncated at 0.

Because the proportional hazards model is not available in `nimble`, `ameras` uses a piecewise constant baseline hazard for Bayesian model averaging. The interval `min(entry), max(exit))` is divided into `prophaz.numints.BMA` subintervals with cutpoints obtained as quantiles of the distribution of event times among cases, and a baseline hazard parameter is estimated for each subinterval.

Value

The output is an object of class `amerasfit` with a component `call` and a component for every method supplied to `methods`. For each method, the output is a list containing

`coefficients`	named vector of model coefficients.
`sd`	named vector of standard deviations.

`CI` data frame with columns `lower` and `upper` giving 95% confidence bounds or credible interval bounds. When the CI method is `"proflik"`, the data frame also has columns `pval.lower` and `pval.upper` (p-values to verify convergence of the root finder) and `iter.lower` and `iter.upper` (number of iterations used by `uniroot`).

`runtime` string with the runtime in seconds.

For RC, ERC, and MCML the following additional output is included:

`vcov` covariance matrix for the full parameter vector.

`convergence.optim` convergence code as returned by `optim`, with 0 indicating convergence and 1 indicating that the maximal number of iterations was reached.

`counts.optim` number of function evaluations used in the model fit returned by `optim`.

`loglik` log-likelihood value at the optimum.

For BMA the output additionally contains:

`samples` MCMC posterior samples, as obtained from `nimble`. This is a list object with `nchains.BMA` components, each a named matrix with the samples from one chain in its rows, with columns corresponding to model parameters.

`Rhat` data frame with two columns, `Rhat` and `n.eff`. The first column contains the Gelman-Rubin statistics $\hat{R} \geq 1$ that can be used to assess convergence of MCMC chains. A value of 1 indicates good convergence and values > 1.05 indicate poor convergence. The effective sample size `n.eff` is a measure of how many independent samples the auto-correlated MCMC samples correspond to. A low effective sample size indicates high correlations and/or poor mixing.

`included.replicates` indices of replicate exposures that were included to obtain the results.

`prophaz.timepoints` for `family="prophaz"`, time points defining the intervals on which the estimated baseline hazards is constant; these are `prophaz.numints.BMA + 1` time points covering the interval `(min(entry), max(exit))`, based on quantiles among observed event times. See Details.

Finally, for FMA the output additionally contains:

`included.samples` the total number of samples included.

`included.replicates` indices of replicate exposures that were included to obtain results. Fits without a valid variance estimate (i.e., non-invertible Hessian or inverse that is not positive definite) or that reach the maximal number of iterations without convergence are filtered out and not used to obtain results.

The class `amerasfit` supports the methods `coef`, `summary`, and traceplot.

References

Roberti, S., Kwon D., Wheeler W., Pfeiffer R. (in preparation). ameras: An R Package to Analyze Multiple Exposure Realizations in Association Studies

Examples

```
data(data, package="ameras")
dosevars <- paste0("V", 1:10)
ameras(data=data, family="gaussian", Y="Y.gaussian", dosevars=dosevars,
M=c("M1", "M2"), X=c("X1","X2"))
```

`data` *Example data*

Description

Data includes outcomes of all six supported types in the appropriately named columns. For proportional hazards regression, the observed exit time is `time` and event status is `event`. For conditional logistic regression, the matched set variable is `setnr`. The data has 10 exposure replicates in columns `V1-V10`.

Examples

```
data(data, package="ameras")

# Display a few rows of the data
data[1:5, ]
```

`traceplot` *Traceplots for MCMC samples*

Description

Produce MCMC traceplots for `amerasfit` objects.

Usage

```
traceplot(object, ...)

## S3 method for class 'amerasfit'
traceplot(object, iter = 5000, Rhat = TRUE, n.eff = TRUE, pdf = FALSE, ...)
```

Arguments

`object`	a `amerasfit` object containing BMA output to be plotted
`iter`	number of iterations to include in the traceplot (defaults to last 5000)
`Rhat`	logical; whether to include R-hat diagnostics in the plot (default TRUE)
`n.eff`	logical; whether to include effective sample size in the plot (default TRUE)
`pdf`	logical; whether to save the output as a PDF (default FALSE)
`...`	additional arguments passed to `MCMCtrace`

Details

Wrapper for `MCMCvis::MCMCtrace` to produce MCMC diagnostic plots. See `?MCMCtrace` for more plotting options that can be provided through `...`.

Value

Traceplots and posterior density plots.

See Also

`MCMCtrace`

Examples

```
data(data, package="ameras")
fit <- ameras(data, methods="BMA", Y="Y.gaussian", dosevars=paste0("V", 1:10))
traceplot(fit)
```

---

transform1 *Exponential parameter transformation*

Description

Applies exponential transformation $f(\theta_i) = \exp(\theta_i) + L_i$ to one or multiple components of parameter vector $\boldsymbol{\theta}$, where L_i are lower limits that can be different for each component

Usage

```
transform1(params, index.t=1:length(params), lowlimit=rep(0,length(index.t)),
  boundcheck=FALSE, boundtol=1e-3, ... )
```

Arguments

`params`	full input parameter vector
`index.t`	indices of parameters to be transformed (default all)
`lowlimit`	lower limits to be applied (default zero), where the k-th component of `lowlimit` is applied to the k-th index in `index.t`
`boundcheck`	whether to produce a warning when any of the transformed parameters are within `boundtol` of `lowlimit`
`boundtol`	tolerance for producing a warning for reaching the boundary
`...`	not used

Value

Transformed parameter vector.

Examples

```
params <- c(.1, .5, 1)
transform1(params, lowlimit=c(0, -1, 1))
```

`transform1.inv` *Inverse of exponential parameter transformation*

Description

Inverse of `transform1` for the purpose of deriving initial values.

Usage

```
transform1.inv(params, index.t=1:length(params), lowlimit=rep(0,length(index.t)), ... )
```

Arguments

`params`	full input parameter vector
`index.t`	indices of parameters to be transformed (default all)
`lowlimit`	lower limits to be applied (default zero), where the k-th component of `lowlimit` is applied to the k-th index in `index.t`
`...`	not used

Value

Transformed parameter vector.

Examples

```
params <- c(.1, .5, 1) # Desired initial values on original scale
transform1.inv(params, lowlimit=c(0, -1, 1)) # Initial values to use on transformed scale
```

`transform1.jacobian` *Jacobian of the exponential parameter transformation*

Description

Computes the Jacobian matrix of transform1. Note that lower limits do not need to be specified as the Jacobian is independent of those

Usage

```
transform1.jacobian(params, index.t=1:length(params), ... )
```

Arguments

params	input parameter vector (before transformation) to evaluate the Jacobian at
index.t	indices of parameters to be transformed (default all)
...	not used

Value

Jacobian matrix.

Examples

```
params <- c(.1, .5, 1)
transform1.jacobian(params)
```

Index

∗ **Bayesian model averaging**
ameras-package, 2
∗ **Monte Carlo Maximum Likelihood**
ameras-package, 2
∗ **data**
data, 8
∗ **exposure replicates**
ameras-package, 2
∗ **exposure uncertainty**
ameras-package, 2
∗ **frequentist model averaging**
ameras-package, 2
∗ **measurement error**
ameras-package, 2
∗ **regression calibration**
ameras-package, 2

ameras, *2*, 3
ameras-package, 2

data, 8

MCMCtrace, *9*

traceplot, *7*, 8
transform1, *5, 6*, 9, *10*
transform1.inv, 10
transform1.jacobian, *6*, 10