

Package: fastglmm (via r-universe)

July 17, 2026

Title Fast (Generalized) Linear Mixed Models via the 'glmm' Rust Kernel

Version 0.1.0

Description Fits linear models, generalized linear models, and (generalized) linear mixed models with a compiled Rust backend (the 'glmm' crate): REML linear mixed models, and Laplace or adaptive Gauss-Hermite quadrature for binomial and Poisson mixed models, using 'lme4'-style formulas. Estimates are validated against 'lme4' and 'MixedModels.jl'. Deliberately scoped to fast fitting - fixed effects, Wald standard errors, and variance components; anything the backend cannot compute honestly (random-effect predictions, likelihood-based criteria, profiling) is a documented error, never a silently different answer.

License GPL (>= 3)

URL <https://github.com/pawlenartowicz/glmm>

BugReports <https://github.com/pawlenartowicz/glmm/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/rextendr/version 0.5.0

SystemRequirements Cargo (Rust's package manager), rustc >= 1.85

Depends R (>= 4.2)

Imports stats

Suggests lme4, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs libclang-dev

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

Date/Publication 2026-07-17 02:11:30 UTC

RemoteUrl <https://github.com/pawlenartowicz/GLMM>

RemoteRef r-universe
RemoteSha 885cc19766e67bc3e9127d85325f653a133b130a

Contents

confint.fastglmm	2
fastglmm	3
fixef	5
formula.fastglmm	6
isSingular	6
ranef	7
sigma.fastglmm	7
summary.fastglmm	8
VarCorr	8
Index	10

confint.fastglmm	<i>Wald confidence intervals</i>
------------------	----------------------------------

Description

Normal-approximation intervals off the full Wald covariance matrix (vcov()). method = "profile" and method = "boot" are not available: the engine has no profiling machinery, and bootstrap needs refitting infrastructure not built yet.

Usage

```
## S3 method for class 'fastglmm'  
confint(object, parm, level = 0.95, method = "Wald", ...)
```

Arguments

- object a [fastglmm](#) fit.
- parm coefficients to include (names or indices; default all).
- level confidence level.
- method only "Wald".
- ... unused.

Value

A matrix with one row per coefficient.

fastglmm	<i>Fit a (generalized) linear or linear mixed model with the glmm Rust kernel</i>
----------	---

Description

One entry point for the whole glmm engine, dispatching on family the way the kernel itself does: OLS (gaussian without random effects), GLM (binomial/poisson/gamma/negative-binomial without random effects), REML LMM (gaussian with random effects), and GLMM (Laplace or adaptive Gauss-Hermite) - there is no lmer/glmer split.

Usage

```
fastglmm(
  formula,
  data,
  family = gaussian(),
  weights = NULL,
  subset = NULL,
  na.action = getOption("na.action"),
  nAGQ = 1L,
  start = NULL,
  wald.se = c("hessian", "rx"),
  dispersion = NULL,
  init.theta = NULL,
  ...
)
```

Arguments

formula	an lme4-style model formula (or a string), e.g. $y \sim t + d + t:d + (1 + t g)$. Bare column names only - see Details.
data	a data.frame (or something coercible) holding every column the formula names.
family	a family object, family function, or string: one of gaussian, binomial (logit/probit), poisson (log), Gamma (log/inverse), "negativebinomial" (log; the shape theta is estimated). inverse.gaussian and binomial("cloglog") are approved for GLMM 0.1.1 and error until the kernel implements them.
weights	optional per-row prior (case) weights - lme4::glmer's weights=. For an aggregated binomial, pass the success proportion as the response and the trial count here (the same model as cbind(successes, failures), whose syntax the parser does not accept).
subset	optional row filter, evaluated in data like lm's subset=. Applied before fitting - row filtering, not parsing.
na.action	how to handle NAs in the model columns (default getOption("na.action"), normally stats::na.omit). NAs must be resolved before the kernel sees the data; an action that leaves them in place (na.pass) is an error.

nAGQ	adaptive Gauss-Hermite node count: an odd integer in 1..=25 (1 = Laplace, the default). More permissive than lme4 in range but see the fallback note in Details.
start	optional warm start, lme4's name and shape: list(beta =, theta =) with theta the random-effect Cholesky vector. Distinct from init.theta, which is the negative-binomial shape.
wald.se	Wald standard-error mode: "hessian" (default) or "rx".
dispersion	Gamma dispersion directive: NULL (estimate via Pearson, the default), "estimate" (same), or a number to hold it fixed. Non-NULL on binomial/poisson would mean quasi-likelihood - GLMM 0.1.1, errors today.
init.theta	negative-binomial shape seed, named for MASS::glm.nb(init.theta=). No kernel hook exists yet to seed the shape search, so any non-NULL value is an error (the default cold start is what runs).
...	intercepted, never silently swallowed: known lme4 arguments (REML, control, verbose, contrasts, offset) raise errors saying why they cannot be honored; unknown names error as unused arguments.

Details

The formula is parsed by the same Rust parser the Python port uses, which accepts **bare column names only**: no I(), poly(), log(x), cbind(), offset(), no ., and no term removal (- 1 / 0 +). Compute transformed columns first and pass them by name. Fixed and random effects support +, :, *, A/B nesting, and (1 + x | g) random-effect terms with a **full** correlation structure - (x || g) and intercept-free RE terms are not fittable by the kernel and raise an error. Contrasts are always treatment coding with the **first factor level** as base; to change the base, relevel() the factor (a contrasts argument is deliberately absent). Character columns are converted to factors with lexicographic level order (as factor() does); a factor's declared level order is honored.

Gamma() **link trap**: R's Gamma() family object defaults to link = "inverse", and a family *object* is honored as given - R semantics win. The string form family = "gamma" uses the glmm default link = "log" instead. The two forms therefore fit different models; choose deliberately.

nAGQ **fallback (louder than lme4)**: nAGQ > 1 is honored on binomial and Poisson mixed models with a single grouping factor and at most 3 random effects per group. Any other shape **warns and falls back to Laplace** (nAGQ = 1) instead of erroring the way lme4::glmer does - the fit you get is a Laplace fit, and the warning is the only notice. This mirrors the Python port so the two ports agree.

Anything the engine cannot do is an error naming the reason - never a silently different model. That includes REML = FALSE (the LMM path is REML-only **by design**), offset(), control=/verbose=, family = inverse.gaussian() and link = "cloglog" (approved for GLMM 0.1.1, not yet in the kernel), and quasi-likelihood dispersion= on binomial/poisson.

Value

An object of class "fastglmm": fixed effects (**fixef**), Wald covariance (vcov()), variance components on the SD/correlation scale (**VarCorr**), converged and singular flags (**isSingular**), plus print(), **summary()**, **confint()** (Wald), nobs(), **formula()** (returns the formula **string**), family(), and model.frame(). Engine-blocked accessors (ranef, predict, fitted, residuals, coef, logLik/AIC, terms) error with the reason.

Examples

```
set.seed(1)
n_g <- 30L
g <- factor(rep(seq_len(n_g), each = 10))
t <- rep(seq(0, 0.9, by = 0.1), n_g)
d <- rbinom(n_g * 10L, 1L, 0.4)
u <- rnorm(n_g, sd = 0.8)[as.integer(g)]
y <- rbinom(n_g * 10L, 1L, plogis(-0.5 + 0.7 * t + 0.4 * d + u))
fit <- fastglmm(y ~ t + d + t:d + (1 | g), data.frame(y, t, d, g),
               family = binomial())
fixef(fit)
VarCorr(fit)
```

fixef	<i>Extract fixed effects</i>
-------	------------------------------

Description

Generic + method. The generic is defined here (masking `lme4::fixef` when both are attached is harmless - S3 dispatch finds the same method).

Usage

```
fixef(object, ...)

## S3 method for class 'fastglmm'
fixef(object, ...)
```

Arguments

object	a fastglmm fit.
...	unused.

Value

Named numeric vector of fixed-effect estimates; aliased (rank-deficient) coefficients are NA, mirroring `lm/lme4`.

formula.fastglm	<i>Formula, family, and model frame accessors</i>
-----------------	---

Description

formula() returns the formula **string** as given: the R side never builds a terms object (the parser is Rust-side, R-port spec section 3), and synthesizing one would disagree with how the model was actually built - which is also why terms() errors.

Usage

```
## S3 method for class 'fastglm'
formula(x, ...)

## S3 method for class 'fastglm'
family(object, ...)

## S3 method for class 'fastglm'
model.frame(formula, ...)
```

Arguments

x, formula, object	
	a fastglm fit.
...	unused.

isSingular	<i>Is the fit singular?</i>
------------	-----------------------------

Description

TRUE iff the fit converged onto the variance-component boundary - the same condition lme4::isSingular reports (the kernel computes it; see glmm::Fit::singular).

Usage

```
isSingular(x, ...)

## S3 method for class 'fastglm'
isSingular(x, ...)
```

Arguments

x	a fastglm fit.
...	unused.

ranef	<i>Random-effect predictions (not available)</i>
-------	--

Description

Errors: conditional modes are computed inside the kernel but not surfaced on `glmm::Fit` yet. The engine spec 2026-07-15-engine-loglik-diagnostics is what lifts this.

Usage

```
ranef(object, ...)

## S3 method for class 'fastglmm'
ranef(object, ...)
```

Arguments

object	a fastglmm fit.
...	unused.

sigma.fastglmm	<i>Residual scale</i>
----------------	-----------------------

Description

The residual standard deviation for **gaussian** fits (sqrt of the kernel's dispersion: $RSS/(n-p)$) for a linear model, matching `summary.lm`'s sigma; the REML residual variance for a mixed model, matching `lme4::sigma`, `sqrt(phi)` for Gamma, and 1 for binomial/poisson/negative-binomial (the scale is fixed, as in `lme4`).

Usage

```
## S3 method for class 'fastglmm'
sigma(object, ...)
```

Arguments

object	a fastglmm fit.
...	unused.

summary.fastglm	<i>Summarize a fastglm fit</i>
-----------------	--------------------------------

Description

The coefficient table carries Wald z statistics and normal-based p-values for all families (the kernel surfaces no residual df, so there is no t). There is deliberately **no** AIC BIC logLik deviance header line: the engine does not surface a comparable log-likelihood yet, and printing a fake one would be worse than omitting it.

Usage

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

Arguments

object	a fastglm fit.
...	unused.

Value

An object of class "summary.fastglm" with a coefficients matrix (Estimate, Std. Error, z value, Pr(>|z|)).

VarCorr	<i>Variance components on the SD/correlation scale</i>
---------	--

Description

Returns one covariance matrix per grouping (declaration order), each with "stddev" and "correlation" attributes, printed lme4-shaped. This is the load-bearing accessor for parameter-recovery use: attr(vc\$g, "stddev") are the tau estimates and attr(vc\$g, "correlation") the rho estimates, straight from the kernel's varcorr (validated against lme4's VarCorr).

Usage

```
VarCorr(x, ...)

## S3 method for class 'fastglm'
VarCorr(x, ...)
```

Arguments

x	a fastglm fit.
...	unused.

Details

For a **gaussian** mixed fit a Residual row is printed after the group components, as in lme4, carrying `sigma()` (the REML residual standard deviation the kernel reports as dispersion).

Value

A list of class `"VarCorr.fastglm"`, one named element per grouping.

Index

`confint()`, [4](#)
`confint.fastglm`, [2](#)

`family.fastglm(formula.fastglm)`, [6](#)
`fastglm`, [2](#), [3](#), [5–8](#)
`fixef`, [4](#), [5](#)
`formula()`, [4](#)
`formula.fastglm`, [6](#)

`isSingular`, [4](#), [6](#)

`model.frame.fastglm`
 (`formula.fastglm`), [6](#)

`ranef`, [7](#)

`sigma.fastglm`, [7](#)
`stats::na.omit`, [3](#)
`summary()`, [4](#)
`summary.fastglm`, [8](#)

`VarCorr`, [4](#), [8](#)