# Package: IBGS (via r-universe)

October 25, 2026

**Type** Package

**Title** Iterated Block Gibbs Sampler for Ultrahigh-Dimensional Variable Selection and Model Averaging

**Version** 1.0.0

**Description** Variable selection for generalized linear models and the Cox proportional-hazards model in ultrahigh dimensions via the iterated block Gibbs sampler (IBGS). The sampler is implemented in C with parallel block screening through 'OpenMP', and supports the gaussian, binomial and poisson families (fitted by least squares or iteratively reweighted least squares) as well as the Cox model for survival analysis (fitted by its Efron partial likelihood), together with the AIC, BIC, AICc and extended BIC model selection criteria.

**License** GPL-3

**Encoding** UTF-8

**Depends** R (>= 3.5.0)

**Imports** graphics, stats

**NeedsCompilation** yes

**Author** Lizhong Chen [aut, cre]

**Maintainer** Lizhong Chen <chen.l@wehi.edu.au>

**Repository** https://cran.r-universe.dev

**Date/Publication** 2026-07-04 08:30:06 UTC

**RemoteUrl** https://github.com/cran/IBGS

**RemoteRef** HEAD

**RemoteSha** 9689a766d4e57fb485e2a5b5108977c4393fb224

## Contents

IBGS-package . . . . . 2
coef.IBGS . . . . . 3
coxGibbs . . . . . 4

coxIBGS . . . . . 5
fitted.IBGS . . . . . 8
glmGibbs . . . . . 9
glmIBGS . . . . . 11
lmeGibbs . . . . . 13
lmeIBGS . . . . . 15
plot.IBGS . . . . . 18
plotAutocorr . . . . . 18
plotGelman . . . . . 19
plotICtrace . . . . . 20
plotMargProb . . . . . 20
plotModelFreq . . . . . 21
predict.IBGS . . . . . 22
print.IBGS . . . . . 23
summary.IBGS . . . . . 24

**Index** **25**

---

`IBGS-package` *Iterated Block Gibbs Sampler for Ultrahigh-Dimensional Variable Selection and Model Averaging*

---

## Description

Variable selection for generalized linear models, the Cox proportional-hazards model and the linear mixed model in ultrahigh dimensions via the iterated block Gibbs sampler (IBGS). The sampler is implemented in C with parallel block screening through OpenMP, and supports the gaussian, binomial and poisson families (fitted by least squares or iteratively reweighted least squares), the Cox model for survival analysis (fitted by its Efron partial likelihood), and the fixed-effect part of a linear mixed model (fitted on whitened data). Candidate models are compared by the AIC, BIC, AICc or extended BIC criterion.

## Details

Each model type comes with two samplers: a *block* sampler that splits the predictors into small groups for ultrahigh dimensions, and a *standalone* (non-block) sampler for problems where the full design is manageable.

| | |
|---|---|
| `glmIBGS` | block sampler for generalized linear models |
| `glmGibbs` | standalone sampler for generalized linear models |
| `coxIBGS` | block sampler for the Cox model |
| `coxGibbs` | standalone sampler for the Cox model |
| `lmeIBGS` | block sampler for the linear mixed model |
| `lmeGibbs` | standalone sampler for the linear mixed model |

Every sampler returns an object of class `"IBGS"` with `print`, `summary`, `coef` and `plot` methods. `predict` and `fitted` form model-averaged predictions, and `plotICtrace`, `plotModelFreq` and `plotMargProb` provide diagnostic plots.

## Author(s)

Lizhong Chen <chen.l@wehi.edu.au>

Maintainer: Lizhong Chen <chen.l@wehi.edu.au>

## References

Chen, L. (2022). *Model selection and averaging by Gibbs sampler with a tropical cyclone seasonal forecasting application*. PhD thesis, University of Melbourne. https://hdl.handle.net/11343/311691

## See Also

glmIBGS, coxIBGS, lmeIBGS, predict.IBGS

## Examples

```
x <- matrix(rnorm(100 * 50), 100, 50)
y <- rowSums(x[, 1:3]) + rnorm(100)
fit <- glmIBGS(y, x, criterion = "BIC")
fit
predict(fit, x[1:5, ])
```

---

`coef.IBGS` *Coefficients of an IBGS fit*

---

## Description

Returns the coefficient vector of one of the retained top models, or the model-averaged coefficient vector (smooth-SIC weights over the best `n.models`, the same weighting `predict.IBGS` uses). The vector is named by the predictors (with `"(Intercept)"` first for the families that carry an intercept).

## Usage

```
## S3 method for class 'IBGS'
coef(object, model = 1, average = FALSE, n.models = object$n.models, ...)
```

## Arguments

| | |
|---|---|
| `object` | an `"IBGS"` result |
| `model` | which retained model (1 = best by criterion), used when `average = FALSE` |
| `average` | average over the top `n.models` instead of a single model |
| `n.models` | number of top models to average when `average = TRUE` |
| `...` | ignored |

## Value

a named numeric vector of coefficients

coxGibbs *The restricted Gibbs sampler for the Cox model*

## Description

The standalone (non-block) Metropolis-within-Gibbs sampler for the Cox proportional-hazards model that searches the model space directly over all predictors, with the model size capped at `max.size`. This is the survival-analysis counterpart of `glmGibbs`: every candidate Cox model is fitted by its Efron partial likelihood in a fast C routine (`cox_gibbs_glm`), and the summary is assembled by refitting the top models in C. Intended for problems where the full design is manageable; use `coxIBGS` in ultrahigh dimensions.

## Usage

```
coxGibbs(
  y,
  status,
  x,
  max.size = ncol(x),
  permute = TRUE,
  n.models = 10,
  threshold = 0.9,
  n.draws = 1000,
  inv.temp = 0.5,
  ebic.gamma = 0.5,
  criterion = c("AIC", "BIC", "AICc", "exBIC"),
  weights = NULL,
  cor.check = NULL,
  start = c("null", "full")
)
```

## Arguments

| | |
|---|---|
| `y` | the event/censoring time (a non-negative numeric vector) |
| `status` | the event indicator: 1 = event (death), 0 = censored |
| `x` | the predictors |
| `max.size` | the maximal number of predictors in a candidate model, default is `ncol(x)` |
| `permute` | the coordinate visiting order within each Gibbs sweep. `TRUE` (default) draws a fresh random permutation each sweep, so every predictor is updated exactly once per sweep (without replacement); `FALSE` uses a fixed in-order systematic sweep. |
| `start` | the initial model for the Gibbs chain(s). `"null"` (default) starts from the empty model (no covariates; the Cox model has no intercept) and grows, avoiding the ill-conditioned full-model start where the criterion (notably AICc) can be degenerate; `"full"` starts from the saturated model (the previous behaviour). |

| | |
|---|---|
| n.models | the number of top selected models, default is 10 |
| threshold | the threshold to select the important predictors, default 0.9 |
| n.draws | the half number of generated samples, default is 1000 |
| inv.temp | the tuning parameter, default is 0.5 |
| ebic.gamma | the parameter for extended BIC, default is 0.5 |
| criterion | the model selection criterion: AIC, BIC, AICc or exBIC |
| weights | optional prior case weights (length nrow(x)); defaults to all ones |
| cor.check | optional correlation threshold; NULL (default) skips the check. A scalar such as 0.9999 runs an O(p^2) scan of the predictors and stops if any pair has absolute correlation above it (naming the offenders), since near-duplicate columns destabilise the fits. |

## Details

coxGibbs runs a single random-permutation Metropolis-within-Gibbs walk over the inclusion indicators of all p predictors, with the model size capped at max.size and no block screening, just as glmGibbs does for the GLM families. The acceptance rule min(1, exp(inv.temp * (IC.current - IC.proposed))), the four information criteria and the model averaging over the top n.models models are as described under glmIBGS; each candidate is fitted by its Efron partial likelihood, the model has no intercept, and the criteria use the number of events as the effective sample size. Use coxIBGS in ultrahigh dimensions.

## Value

an object of class "IBGS" summarising the search, with components marginal.prob (the marginal inclusion probability of each predictor), selected.vars (the names of the predictors above threshold), var.names, coef (the coefficient matrix of the top n.models models), model.ic and model.freq (the criterion and visit frequency of each top model), ic.trace (the criterion at every generation) and criterion (the criterion name). Has print, summary, coef and plot methods.

## Examples

```
x <- matrix(rnorm(2000), ncol = 10)
eta <- x[, 1:3] %*% c(1, -1, 0.5)
time <- rexp(200, exp(eta))
status <- rbinom(200, 1, 0.8)
m.s <- coxGibbs(time, status, x, criterion = "BIC")
```

---

coxIBGS *The iterated block Gibbs sampler for the Cox model*

---

## Description

Variable selection for the Cox proportional-hazards model in ultrahigh dimensions. This is the survival-analysis counterpart of glmIBGS: the whole search (refinement iterations, random block screening and the final long run) is performed in a single multicore C routine (cox_ibgs_glm), and this R function is a thin wrapper that assembles the summary (refitting the top models in C). Each candidate model is fitted by its Efron partial likelihood; the Cox model has no intercept, and the information criteria use the number of events as the effective sample size.

## Usage

```
coxIBGS(
  y,
  status,
  x,
  n.refine = 3,
  n.models = 10,
  block.size = 30,
  n.keep = 20,
  threshold = 0.9,
  permute = TRUE,
  n.draws = 250,
  inv.temp = 0.5,
  ebic.gamma = 0.5,
  criterion = c("AIC", "BIC", "AICc", "exBIC"),
  weights = NULL,
  n.cores = 1L,
  cor.check = NULL,
  start = c("null", "full")
)
```

## Arguments

| | |
|---|---|
| y | the event/censoring time (a non-negative numeric vector) |
| status | the event indicator: 1 = event (death), 0 = censored |
| x | the predictors |
| n.refine | the number of refinement iterations, default is 3 |
| n.models | the number of top selected models, default is 10 |
| block.size | the number of predictors in small groups, default is 30 |
| n.keep | the number of selected predictors in the first step, default 20 |
| threshold | the threshold to select the important predictors, default 0.9 |
| permute | the coordinate visiting order within each Gibbs sweep. TRUE (default) draws a fresh random permutation each sweep, so every predictor is updated exactly once per sweep (without replacement); FALSE uses a fixed in-order systematic sweep. |

| | |
|---|---|
| start | the initial model for the Gibbs chain(s). "null" (default) starts from the empty model (no covariates; the Cox model has no intercept) and grows, avoiding the ill-conditioned full-model start where the criterion (notably AICc) can be degenerate; "full" starts from the saturated model (the previous behaviour). |
| n.draws | the half number of generated samples, default is 250 |
| inv.temp | the tuning parameter, default is 0.5 |
| ebic.gamma | the parameter for extended BIC, default is 0.5 |
| criterion | the model selection criterion: AIC, BIC, AICc or exBIC |
| weights | optional prior case weights (length nrow(x)); defaults to all ones |
| n.cores | the number of OpenMP threads for the block screening (0 = use all available cores), default is 1 (serial) |
| cor.check | optional correlation threshold; NULL (default) skips the check. A scalar such as 0.9999 runs an O(p^2) scan of the predictors and stops if any pair has absolute correlation above it (naming the offenders), since near-duplicate columns destabilise the fits. |

## Details

coxIBGS applies the same iterated block Gibbs algorithm as glmIBGS – the screen / select / threshold refinement loop, the final long Gibbs run, the Metropolis-within-Gibbs acceptance rule min(1, exp(inv.temp * (IC.current - IC.proposed))), the four information criteria and the model averaging over the top n.models models – but for the Cox proportional-hazards model rather than a GLM. Each candidate model is fitted in C by its Efron partial likelihood; the Cox model carries no intercept, and the information criteria use the number of events (not the sample size n) as the effective sample size. See glmIBGS for the screening, sampler and criteria details.

## Value

an object of class "IBGS" summarising the search, with components marginal.prob (the marginal inclusion probability of each predictor), selected.vars (the names of the predictors above threshold), var.names, coef (the coefficient matrix of the top n.models models), model.ic and model.freq (the criterion and visit frequency of each top model), ic.trace (the criterion at every generation) and criterion (the criterion name). Has print, summary, coef and plot methods.

## Examples

```
x <- matrix(rnorm(60 * 20), 60, 20)
time <- rexp(60, exp(rowSums(x[, 1:3])))
status <- rbinom(60, 1, 0.7)
fit <- coxIBGS(time, status, x, criterion = "BIC")
fit
```

`fitted.IBGS` *Fitted values of an IBGS fit*

## Description

The model-averaged fitted values on the training data, on the response (default) or link scale. The averaged training linear predictor (the `linear.predictors` component) is computed once at fit time and stored on the object, so this is cheap even in high dimensions. Equivalent to `predict(object)` with no new data.

## Usage

```
## S3 method for class 'IBGS'
fitted(object, type = c("response", "link"), ...)
```

## Arguments

| | |
|---|---|
| `object` | an `"IBGS"` result |
| `type` | `"response"` (the default, response scale) or `"link"` (the linear-predictor scale). |
| `...` | ignored |

## Value

a numeric vector of model-averaged fitted values on the training data (length equal to the number of training observations).

## See Also

`predict.IBGS`, `coef.IBGS`

## Examples

```
x <- matrix(rnorm(1000), ncol = 10)
y <- rowSums(x[, 1:5]) + rnorm(100)
fit <- glmGibbs(y, x, criterion = "BIC")
head(fitted(fit))
```

`glmGibbs` *The restricted Gibbs sampler (linear model)*

## Description

The standalone (non-block) Metropolis-within-Gibbs sampler that searches the model space directly over all predictors, with the model size capped at `max.size`. Every candidate linear model is fitted by ordinary least squares in a fast C routine (`gibbs_sampler_gaussian`); the summary is assembled by refitting the top models in C to recover their coefficients. Unlike `glmIBGS`, this sampler does not split the predictors into blocks and is intended for problems where the full design is manageable.

## Usage

```
glmGibbs(
  y,
  x,
  max.size = ncol(x),
  permute = TRUE,
  n.models = 10,
  threshold = 0.9,
  n.draws = 1000,
  inv.temp = 0.5,
  ebic.gamma = 0.5,
  criterion = c("AIC", "BIC", "AICc", "exBIC"),
  family = c("gaussian", "binomial", "poisson"),
  weights = NULL,
  cor.check = NULL,
  start = c("null", "full"),
  fast = TRUE
)
```

## Arguments

| | |
|---|---|
| `y` | the response variable |
| `x` | the predictors |
| `max.size` | the maximal number of predictors in a candidate model, default is `ncol(x)` |
| `permute` | the coordinate visiting order within each Gibbs sweep. `TRUE` (default) draws a fresh random permutation each sweep, so every predictor is updated exactly once per sweep (without replacement); `FALSE` uses a fixed in-order systematic sweep. |
| `start` | the initial model for the Gibbs chain(s). `"null"` (default) starts from the empty model (intercept only; no covariates for the Cox model) and grows, avoiding the ill-conditioned full-model start where the criterion (notably AICc) can be degenerate; `"full"` starts from the saturated model (the previous behaviour). |

| | |
|---|---|
| fast | for the binomial/poisson families only, a speed/approximation trade-off in the iterative (IRLS) fit. FALSE fits every candidate model to full convergence. TRUE (default) scores each single-coordinate proposal with one warm-started IRLS step and re-fits only the accepted models to convergence, so the recorded criteria stay exact but the accept/reject decision uses an approximate score; much faster when the iterative fit dominates the run. Ignored for the gaussian family (a direct, non-iterative fit). |
| n.models | the number of top selected models, default is 10 |
| threshold | the threshold to select the important predictors, default 0.9 |
| n.draws | the half number of generated samples, default is 1000 |
| inv.temp | the tuning parameter, default is 0.5 |
| ebic.gamma | the parameter for extended BIC, default is 0.5 |
| criterion | the selected model selection criterion from AIC, AICc, BIC and exBIC |
| family | the model family: "gaussian", "binomial" or "poisson" |
| weights | optional prior weights (length nrow(x)); as in glm. For the binomial family y may be a fraction (proportion of successes) with weights the number of trials. Defaults to all ones. |
| cor.check | optional correlation threshold; NULL (default) skips the check. A scalar such as 0.9999 runs an O(p^2) scan of the predictors and stops if any pair has absolute correlation above it (naming the offenders), since near-duplicate columns destabilise the fits. |

## Details

glmGibbs runs a single random-permutation Metropolis-within-Gibbs walk directly over the inclusion indicators of all p predictors, with the model size capped at max.size; there is no block screening or refinement. The flip acceptance rule min(1, exp(inv.temp * (IC.current - IC.proposed))), the four information criteria (AIC, BIC, AICc and exBIC) and the model averaging over the top n.models models are exactly as described under glmIBGS, and each candidate is fitted in the same way (gaussian by ordinary least squares, binomial and poisson by IRLS). Because every candidate fit may use all p predictors, this sampler is intended for problems where the full design is manageable (p not much larger than n); use glmIBGS in ultrahigh dimensions.

## Value

an object of class "IBGS" summarising the search, with components marginal.prob (the marginal inclusion probability of each predictor), selected.vars (the names of the predictors above threshold), var.names, coef (the coefficient matrix of the top n.models models), model.ic and model.freq (the criterion and visit frequency of each top model), ic.trace (the criterion at every generation) and criterion (the criterion name). Has print, summary, coef and plot methods.

## Examples

```
x <- matrix(rnorm(1000), ncol = 10)
y <- rowSums(x[,1:5]) + rnorm(100)
m.s <- glmGibbs(y, x, criterion = "BIC")
```

`glmIBGS` *The iterated block Gibbs sampler algorithm*

## Description

Variable selection for the generalized linear model in ultrahigh dimensions. The whole search – the refinement iterations, the random block screening, and the final long run – is performed in a single multicore C routine (`ibgs_glm`); this R function is a thin wrapper that assembles the result summary (refitting the top models in C to recover their coefficients). The gaussian family is fitted by ordinary least squares; the binomial and poisson families by IRLS.

## Usage

```
glmIBGS(
  y,
  x,
  n.refine = 3,
  n.models = 10,
  block.size = 30,
  n.keep = 20,
  threshold = 0.9,
  permute = TRUE,
  n.draws = 250,
  inv.temp = 0.5,
  ebic.gamma = 0.5,
  criterion = c("AIC", "BIC", "AICc", "exBIC"),
  family = c("gaussian", "binomial", "poisson"),
  weights = NULL,
  n.cores = 1L,
  cor.check = NULL,
  start = c("null", "full"),
  fast = TRUE
)
```

## Arguments

| | |
|---|---|
| `y` | the response variable |
| `x` | the predictors |
| `n.refine` | the number of refinement iterations, default is 3 |
| `n.models` | the number of top selected models |
| `block.size` | the number of predictors in small groups, default is 30 |
| `n.keep` | the number of selected predictors in first step, default is 20 |
| `threshold` | the threshold to select the important predictors in second step, default is 0.9 |

| | |
|---|---|
| permute | the coordinate visiting order within each Gibbs sweep. TRUE (default) draws a fresh random permutation each sweep, so every predictor is updated exactly once per sweep (without replacement); FALSE uses a fixed in-order systematic sweep. |
| start | the initial model for the Gibbs chain(s). "null" (default) starts from the empty model (intercept only; no covariates for the Cox model) and grows, avoiding the ill-conditioned full-model start where the criterion (notably AICc) can be degenerate; "full" starts from the saturated model (the previous behaviour). |
| fast | for the binomial/poisson families only, a speed/approximation trade-off in the iterative (IRLS) fit. FALSE fits every candidate model to full convergence. TRUE (default) scores each single-coordinate proposal with one warm-started IRLS step and re-fits only the accepted models to convergence, so the recorded criteria stay exact but the accept/reject decision uses an approximate score; much faster when the iterative fit dominates the run. Ignored for the gaussian family (a direct, non-iterative fit). |
| n.draws | the half number of generated samples, default is 250 |
| inv.temp | the tuning parameter, default is 0.5 |
| ebic.gamma | the parameter for extended BIC, default is 0.5 |
| criterion | the selected model selection criterion from AIC, AICc, BIC and exBIC |
| family | the model family: "gaussian", "binomial" or "poisson" |
| weights | optional prior weights (length nrow(x)); as in glm. For the binomial family y may be a fraction (proportion of successes) with weights the number of trials. Defaults to all ones. |
| n.cores | the number of OpenMP threads for the block screening (0 = use all available cores), default is 1 (serial) |
| cor.check | optional correlation threshold; NULL (default) skips the check. A scalar such as 0.9999 runs an O(p^2) scan of the predictors and stops if any pair has absolute correlation above it (naming the offenders), since near-duplicate columns destabilise the fits. |

## Details

In ultrahigh dimensions the number of predictors p can far exceed the sample size n, so a single Gibbs sampler over all p inclusion indicators is both slow (each candidate fit can use at most n columns) and noisy. IBGS instead alternates *screening* and *refinement* to grow a small, stable set of important predictors:

- *Screen.* Randomly partition the not-yet-important predictors into blocks of size block.size (capped at n minus the current important set, so every block design is estimable). An independent within-block Gibbs sampler is run on each block – always including the important set – giving a marginal inclusion probability for every predictor in the block.
- *Select.* Keep the top n.keep predictors by inclusion probability, union them with the important set, and run one combined Gibbs sampler over that union.
- *Threshold.* Promote every predictor whose combined inclusion probability exceeds threshold into the important set.

These steps are repeated for `n.refine` refinement rounds; a final screen is then followed by a long Gibbs run over the final candidate set, which produces the recorded indicator samples, their information criteria, and the marginal inclusion probabilities returned to R.

Each Gibbs sampler is a random-permutation Metropolis-within-Gibbs walk over the model indicator vector: each sweep visits every coordinate once in a fresh random order, flipping it and accepting the flip with probability `min(1, exp(inv.temp * (IC.current - IC.proposed)))`. Because a lower criterion is better, a flip that lowers the criterion is always accepted and a worsening flip is accepted with a probability that decays in the increase; the tuning parameter `inv.temp` controls how greedy the walk is. The chain's stationary distribution favours low-criterion models and its visit frequencies estimate the marginal inclusion probabilities. Every candidate model is fitted in C – the gaussian family by ordinary least squares (a Cholesky solve on the active sub-block of a precomputed Gram matrix), the binomial and poisson families by iteratively reweighted least squares.

Models are compared by one of four information criteria, each of the form `-2logLik + penalty`: `AIC = -2logLik + 2 npar`, `BIC = -2logLik + log(n) npar`, `AICc = AIC + 2 npar (npar + 1) / (n - npar - 1)` (a small-sample AIC), and `exBIC = BIC + 2 ebic.gamma npred log(p0)` (the extended BIC of Chen and Chen, 2008, whose extra term penalises the size `p0` of the candidate pool and is essential in ultrahigh dimensions where BIC under-penalises). Here `npar` counts the estimated parameters and `npred` the predictors excluding the intercept.

The returned object summarises the search: the top `n.models` distinct models are refit in C to recover their coefficients, and averaged coefficients and predictions combine them with smooth-criterion weights (see `coef.IBGS` and `predict.IBGS`). The block screening is parallelised with `n.cores` OpenMP threads (one thread per block); the random block assignment and the per-block seeds are drawn serially from R's generator, so the result is reproducible and does not depend on the number of threads.

## Value

an object of class `"IBGS"` summarising the search, with components `marginal.prob` (the marginal inclusion probability of each predictor), `selected.vars` (the names of the predictors above `threshold`), `var.names`, `coef` (the coefficient matrix of the top `n.models` models), `model.ic` and `model.freq` (the criterion and visit frequency of each top model), `ic.trace` (the criterion at every generation) and `criterion` (the criterion name). Has `print`, `summary`, `coef` and `plot` methods.

## Examples

```
x <- matrix(rnorm(100 * 50), 100, 50)
y <- rowSums(x[, 1:3]) + rnorm(100)
fit <- glmIBGS(y, x, criterion = "BIC")
fit
```

---

`lmeGibbs` *The restricted Gibbs sampler for the linear mixed model (fixed-effect part)*

---

## Description

The standalone (non-block) Metropolis-within-Gibbs sampler that selects the fixed effects of a linear mixed model while holding the random part fixed. With the random structure held, the marginal model is `y ~ N(X beta, V)` with `V` fixed; whitening by `V = L Lt` reduces each candidate fit to ordinary least squares on the whitened data, performed in a fast C routine. The variance component(s) are estimated once (random-intercept case) or supplied, and held fixed across all candidate models. This is the mixed-model counterpart of `glmGibbs`; use `lmeIBGS` in ultrahigh dimensions.

## Usage

```
lmeGibbs(
  y,
  x,
  group = NULL,
  Z = NULL,
  varcomp = NULL,
  V = NULL,
  max.size = ncol(x),
  permute = TRUE,
  n.models = 10,
  threshold = 0.9,
  n.draws = 1000,
  inv.temp = 0.5,
  ebic.gamma = 0.5,
  criterion = c("AIC", "BIC", "AICc", "exBIC"),
  cor.check = NULL,
  start = c("null", "full")
)
```

## Arguments

| | |
|---|---|
| `y` | the response variable |
| `x` | the fixed-effect predictors |
| `group` | a grouping factor (length `nrow(x)`) for a random intercept |
| `Z` | a random-effects design matrix (general random part) |
| `varcomp` | a list `list(sigma2b, sigma2e)` of variance components used with `Z` |
| `V` | the marginal covariance `V` (n x n), supplied directly |
| `max.size` | the maximal number of fixed predictors in a candidate model, default is `ncol(x)` |
| `permute` | the coordinate visiting order within each Gibbs sweep. `TRUE` (default) draws a fresh random permutation each sweep, so every predictor is updated exactly once per sweep (without replacement); `FALSE` uses a fixed in-order systematic sweep. |
| `start` | the initial model for the Gibbs chain(s). `"null"` (default) starts from the empty model (intercept only) and grows, avoiding the ill-conditioned full-model start where the criterion (notably AICc) can be degenerate; `"full"` starts from the saturated model (the previous behaviour). |

| | |
|---|---|
| n.models | the number of top selected models, default is 10 |
| threshold | the threshold to select the important predictors, default 0.9 |
| n.draws | the half number of generated samples, default is 1000 |
| inv.temp | the tuning parameter, default is 0.5 |
| ebic.gamma | the parameter for extended BIC, default is 0.5 |
| criterion | the model selection criterion: AIC, BIC, AICc or exBIC |
| cor.check | optional correlation threshold; NULL (default) skips the check. A scalar such as 0.9999 stops if any pair of predictors has absolute correlation above it. |

## Details

lmeGibbs runs a single random-permutation Metropolis-within-Gibbs walk over the inclusion indicators of all p fixed-effect predictors, with the model size capped at max.size and no block screening, just as glmGibbs does for the GLM families – but on *whitened* data. With the random part held fixed the marginal model is y ~ N(X beta, V) with V fixed; whitening by V = L Lt reduces each candidate fit to ordinary least squares. The acceptance rule min(1, exp(inv.temp * (IC.current - IC.proposed))), the four information criteria and the model averaging over the top n.models models are as described under glmIBGS; the variance component(s) are estimated once (random-intercept case, by REML) or supplied, and held fixed. Use lmeIBGS in ultrahigh dimensions.

Specify the random part with exactly one of: group (a grouping factor for a random intercept; its variance ratio is estimated once by REML), Z + varcomp, or a marginal covariance V.

## Value

an object of class "IBGS" summarising the search, with components marginal.prob (the marginal inclusion probability of each predictor), selected.vars (the names of the predictors above threshold), var.names, coef (the coefficient matrix of the top n.models models), model.ic and model.freq (the criterion and visit frequency of each top model), ic.trace (the criterion at every generation), criterion (the criterion name) and, when the random part is given via group or Z + varcomp, re (the random-effect BLUPs used for conditional prediction; a directly supplied V gives the fixed-effect fit without re). Has print, summary, coef and plot methods.

## Examples

```
x <- matrix(rnorm(1000), ncol = 10)
g <- gl(10, 10)
y <- rowSums(x[, 1:5]) + rnorm(10)[g] + rnorm(100)
fit <- lmeGibbs(y, x, group = g, criterion = "BIC")
```

---

lmeIBGS *The iterated block Gibbs sampler for the linear mixed model (fixed-effect part)*

---

## Description

Fixed-effect variable selection for a linear mixed model in ultrahigh dimensions, holding the random part fixed. This is the mixed-model counterpart of glmIBGS: the whole search (refinement iterations, random block screening and the final long run) runs in a single multicore C routine on the whitened data. With the random structure held the marginal model is `y ~ N(X beta, V)` with `V` fixed; whitening by `V = L Lt` reduces each candidate fit to ordinary least squares. The variance component(s) are estimated once (random-intercept case, by REML from the largest estimable fixed model) or supplied, and held fixed.

## Usage

```
lmeIBGS(
  y,
  x,
  group = NULL,
  Z = NULL,
  varcomp = NULL,
  V = NULL,
  n.refine = 3,
  n.models = 10,
  block.size = 30,
  n.keep = 20,
  threshold = 0.9,
  permute = TRUE,
  n.draws = 250,
  inv.temp = 0.5,
  ebic.gamma = 0.5,
  criterion = c("AIC", "BIC", "AICc", "exBIC"),
  n.cores = 1L,
  cor.check = NULL,
  start = c("null", "full")
)
```

## Arguments

| | |
|---|---|
| `y` | the response variable |
| `x` | the fixed-effect predictors |
| `group` | a grouping factor (length `nrow(x)`) for a random intercept |
| `Z` | a random-effects design matrix (general random part) |
| `varcomp` | a list `list(sigma2b, sigma2e)` of variance components used with `Z` |
| `V` | the marginal covariance `V` (n x n), supplied directly |
| `n.refine` | the number of refinement iterations, default is 3 |
| `n.models` | the number of top selected models, default is 10 |
| `block.size` | the number of predictors in small groups, default is 30 |
| `n.keep` | the number of selected predictors in the first step, default 20 |

| | |
|---|---|
| threshold | the threshold to select the important predictors, default 0.9 |
| permute | the coordinate visiting order within each Gibbs sweep. TRUE (default) draws a fresh random permutation each sweep, so every predictor is updated exactly once per sweep (without replacement); FALSE uses a fixed in-order systematic sweep. |
| start | the initial model for the Gibbs chain(s). "null" (default) starts from the empty model (intercept only) and grows, avoiding the ill-conditioned full-model start where the criterion (notably AICc) can be degenerate; "full" starts from the saturated model (the previous behaviour). |
| n.draws | the half number of generated samples, default is 250 |
| inv.temp | the tuning parameter, default is 0.5 |
| ebic.gamma | the parameter for extended BIC, default is 0.5 |
| criterion | the model selection criterion: AIC, BIC, AICc or exBIC |
| n.cores | the number of OpenMP threads for the block screening (0 = use all available cores), default is 1 (serial) |
| cor.check | optional correlation threshold; NULL (default) skips the check. A scalar such as 0.9999 stops if any pair of predictors has absolute correlation above it. |

## Details

lmeIBGS runs the same iterated block Gibbs algorithm as glmIBGS – the screen / select / threshold refinement loop, the final long Gibbs run, the acceptance rule min(1, exp(inv.temp * (IC.current - IC.proposed))), the four information criteria and the model averaging over the top n.models models – but on *whitened* data. With the random part held fixed the marginal model is y ~ N(X beta, V) with V fixed; whitening by V = L Lt (left-multiplying y and X by the inverse root) reduces each candidate fit to ordinary least squares, so the C sampler is the gaussian one applied to the whitened design. The variance component(s) are estimated once – for a random intercept by REML from the largest estimable fixed model – or supplied directly, and then held fixed across all candidate models. See glmIBGS for the screening, sampler and criteria details.

Specify the random part with exactly one of: group (a grouping factor for a random intercept), Z + varcomp, or a marginal covariance V.

## Value

an object of class "IBGS" summarising the search, with components marginal.prob (the marginal inclusion probability of each predictor), selected.vars (the names of the predictors above threshold), var.names, coef (the coefficient matrix of the top n.models models), model.ic and model.freq (the criterion and visit frequency of each top model), ic.trace (the criterion at every generation), criterion (the criterion name) and, when the random part is given via group or Z + varcomp, re (the random-effect BLUPs used for conditional prediction; a directly supplied V gives the fixed-effect fit without re). Has print, summary, coef and plot methods.

## Examples

```
x <- matrix(rnorm(100 * 50), 100, 50)
g <- gl(10, 10)
y <- rowSums(x[, 1:3]) + rnorm(10)[g] + rnorm(100)
```

```
fit <- lmeIBGS(y, x, group = g, criterion = "BIC")
fit
```

---

`plot.IBGS` *Diagnostic plots for an IBGS fit*

---

## Description

Dispatches to the diagnostic plots: the trace of the criterion sequence (`plotICtrace`), the marginal inclusion probabilities (`plotMargProb`), the top-model visit frequencies (`plotModelFreq`), the Gelman-Rubin shrink factor (`plotGelman`), and the criterion-trace autocorrelation (`plotAutocorr`). Several may be requested at once, in which case they are drawn in a multi-panel layout.

## Usage

```
## S3 method for class 'IBGS'
plot(
  x,
  which = c("ictrace", "margprob", "modelfreq", "gelman", "autocorr"),
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | an `"IBGS"` result |
| `which` | one or more of `"ictrace"`, `"margprob"`, `"modelfreq"`, `"gelman"`, `"autocorr"`; default is all five |
| `...` | further graphical parameters, forwarded only when a single panel is requested |

## Value

`x`, invisibly

---

`plotAutocorr` *Autocorrelation plot for the criterion trace*

---

## Description

Draws the lagged autocorrelation of the criterion trace (`result$convergence$autocorr`) as a stick plot. A trace that mixes well decays to zero quickly; persistent high autocorrelation signals a slowly-mixing chain and a small effective sample size.

## Usage

```
plotAutocorr(result, col = NULL, ...)
```

### Arguments

| | |
|---|---|
| result | an "IBGS" result from one of the samplers |
| col | colour of the sticks, default from the package palette |
| ... | further graphical parameters passed to plot |

### Details

The autocorrelations are computed in C with no dependency on the **coda** package, matching stats::acf applied to the criterion trace.

### Value

invisibly, the named vector of autocorrelations (named by lag)

---

## plotGelman *Gelman-Rubin shrink-factor plot for the criterion trace*

---

### Description

Plots the evolving Gelman-Rubin potential scale reduction factor (R-hat) of the criterion trace as the run lengthens, computed from the single chain split into equal contiguous segments (result$convergence$shrink). The median and the 97.5% upper limit are drawn together with a horizontal guide at 1 (perfect mixing) and at 1.1 (the usual convergence rule of thumb): once both curves settle below about 1.1 the sampler has stabilised.

### Usage

```
plotGelman(result, col = NULL, legend = TRUE, ...)
```

### Arguments

| | |
|---|---|
| result | an "IBGS" result from one of the samplers |
| col | colour of the median R-hat curve, default from the package palette |
| legend | draw an explanatory legend, default TRUE |
| ... | further graphical parameters passed to plot |

### Details

The diagnostic is computed in C with no dependency on the **coda** package, but follows the same definition as coda::gelman.diag / coda::gelman.plot applied to the criterion trace split into segments (split-R-hat).

### Value

invisibly, the data.frame of shrink-factor values

## plotICtrace *Trace plot of the information-criterion sequence*

### Description

Plots the model selection criterion at each Gibbs generation. The best (lowest) criterion value reached is highlighted and a horizontal guide marks the best retained model's criterion, so the minimal information criterion stands out.

### Usage

```
plotICtrace(
  result,
  col = NULL,
  highlight = NULL,
  running.min = FALSE,
  legend = TRUE,
  ...
)
```

### Arguments

| | |
|---|---|
| result | an "IBGS" result from one of the samplers |
| col | colour of the criterion trace, default from the package palette (grey) |
| highlight | colour marking the best (lowest) value, default orange |
| running.min | add a best-so-far (running minimum) trace, default FALSE |
| legend | draw an explanatory legend, default TRUE |
| ... | further graphical parameters passed to plot |

### Value

invisibly, a list with the best value and the generation that reached it

## plotMargProb *Plot the marginal inclusion probability of the top covariates*

### Description

Draws a dot-and-whisker plot of the n.vars covariates with the highest marginal inclusion probability, sorted descending. Each point carries an error bar of plus/minus one Monte-Carlo standard error of the inclusion proportion, $se = \sqrt{p(1-p)/ESS}$, where ESS is the effective sample size of the criterion trace (result$convergence$ess); the bars are clamped to $[0, 1]$. The selected predictors (result$selected.vars) are highlighted and the rest are muted, so the selected set stands out. With horizontal = TRUE the predictors run down the y-axis (names read horizontally on the left); otherwise they run along the x-axis (the default).

## Usage

```
plotMargProb(
  result,
  n.vars = 20,
  col = NULL,
  horizontal = FALSE,
  lwd = 2,
  side = 1,
  line = 0.25,
  las = 2,
  cex = 1
)
```

## Arguments

| | |
|---|---|
| result | an "IBGS" result from one of the samplers |
| n.vars | the number of top covariates to show, default 20 |
| col | colour of the non-selected points, default from the palette |
| horizontal | lay the predictors down the y-axis (names read horizontally on the left) instead of along the x-axis, default FALSE |
| lwd | error-bar line width |
| side | margin side for the predictor labels (vertical layout; see mtext) |
| line | margin line for the labels |
| las | label orientation (vertical layout; 2 = perpendicular) |
| cex | label character expansion |

## Value

invisibly, the named vector of plotted inclusion probabilities

---

# plotModelFreq *Plot the visit frequency of the top selected models*

---

## Description

Draws a bar for each of the top n.models models, showing how often the sampler settled on it: the relative frequency with which the Gibbs chain visited that model (result$model.freq). The bars are ordered best-first (the model with the lowest information criterion on the left, highlighted), and each bar is annotated with its criterion value. Tall leading bars indicate a search that concentrated sharply on a few models.

### Usage

```
plotModelFreq(
  result,
  n.models = result$n.models,
  col = NULL,
  annotate = TRUE,
  cumulative = FALSE,
  horizontal = FALSE,
  ...
)
```

### Arguments

| | |
|---|---|
| result | an "IBGS" result from one of the samplers |
| n.models | the number of top models to show, default is all retained |
| col | bar colour for the non-best models, default from the palette |
| annotate | write each bar's criterion value above it, default TRUE |
| cumulative | overlay the cumulative visit frequency, default FALSE |
| horizontal | draw horizontal bars (model rank down the y-axis) instead of vertical bars, default FALSE |
| ... | further graphical parameters passed to barplot |

### Value

invisibly, the vector of plotted relative model frequencies

---

## predict.IBGS *Predict method for an IBGS fit*

---

### Description

Model-averaged prediction (the default) or single-model prediction on new data. The top n.models models are averaged on the linear-predictor (link) scale with smooth-SIC weights (proportional to exp(-inv.temp * (IC - min IC))) and the family inverse link is applied once; averaging before the inverse link keeps binomial predictions in the unit interval and poisson/cox predictions positive. With newdata = NULL the model-averaged fitted values stored on the object at fit time are returned.

### Usage

```
## S3 method for class 'IBGS'
predict(object, newdata = NULL, type = c("response", "link"),
  average = TRUE, n.models = object$n.models, model = 1,
  group.new = NULL, Z.new = NULL, ...)
```

### Arguments

| | |
|---|---|
| object | an "IBGS" result from glmGibbs, glmIBGS, coxGibbs, coxIBGS, lmeGibbs or lmeIBGS |
| newdata | new predictor data (a matrix or data frame with the original predictor columns). When NULL the model-averaged fitted values on the training data are returned (see fitted.IBGS). |
| type | "response" (the default, response scale) or "link" (the linear-predictor scale). |
| average | average over the top n.models models (the default, Bayesian-style model averaging) instead of using a single model. |
| n.models | the number of top models to average over when average = TRUE. |
| model | which retained model to use when average = FALSE (1 = the best by criterion). |
| group.new | for an lme fit, the grouping factor of the new data; when supplied the matching model-averaged random-intercept BLUP is added (conditional prediction). Unseen groups contribute 0 (the marginal mean). |
| Z.new | for a general lme fit, the random-effects design of the new data; when supplied Z.new %*% b_hat is added. |
| ... | ignored |

### Value

a numeric vector of predictions: the response scale for the GLM families (probabilities in [0, 1] for binomial), the relative risk exp(x*beta) for cox, and the fixed-effect mean – optionally plus the random-effect BLUP – for lme.

### See Also

fitted.IBGS, coef.IBGS

### Examples

```
x <- matrix(rnorm(1000), ncol = 10)
y <- rowSums(x[, 1:5]) + rnorm(100)
fit <- glmGibbs(y, x, criterion = "BIC")
predict(fit, x[1:5, ])
```

---

## print.IBGS *Print a concise summary of an IBGS fit*

---

### Description

Print a concise summary of an IBGS fit

### Usage

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

### Arguments

| | |
|---|---|
| x | an "IBGS" result from one of the samplers |
| ... | ignored |

### Value

x, invisibly

---

`summary.IBGS` *Tabular summary of an IBGS fit*

---

### Description

Tabular summary of an IBGS fit

### Usage

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

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

### Arguments

| | |
|---|---|
| object | an "IBGS" result from one of the samplers |
| ... | ignored |
| x | a "summary.IBGS" object |

### Value

invisibly, an object of class "summary.IBGS" with the selected variable table, the top-model table and the convergence diagnostics of the criterion trace (Gelman-Rubin R-hat, Geweke z, effective sample size and lag-1 autocorrelation), which print displays as a "Convergence diagnostics" block

# Index


∗ **package**
IBGS-package, 2

barplot, *22*

coef, *2*
coef.IBGS, 3, *8*, *13*, *23*
coxGibbs, *2*, 4, *23*
coxIBGS, *2–4*, 5, *5*, *23*

fitted, *2*
fitted.IBGS, 8, *23*

glm, *10*, *12*
glmGibbs, *2*, *4, 5*, 9, *14, 15*, *23*
glmIBGS, *2, 3*, *5–7*, *9, 10*, 11, *15–17*, *23*

IBGS *(*IBGS-package*)*, 2
IBGS-package, 2

lmeGibbs, *2*, 13, *23*
lmeIBGS, *2, 3*, *14*, 15, *15*, *23*

mtext, *21*

plot, *2*, *19, 20*
plot.IBGS, 18
plotAutocorr, 18, *18*
plotGelman, *18*, 19
plotICtrace, *2*, *18*, 20
plotMargProb, *2*, *18*, 20
plotModelFreq, *2*, *18*, 21
predict, *2*
predict.IBGS, *3*, *8*, *13*, 22
print, *2*
print.IBGS, 23
print.summary.IBGS *(*summary.IBGS*)*, 24

summary, *2*
summary.IBGS, 24