# Package 'glmmPen'

August 25, 2021

**Type** Package

**Title** High Dimensional Penalized Generalized Linear Mixed Models (pGLMM)

**Version** 1.5.1.5

**Date** 2018-10-26

**Maintainer** Hillary Heiling <hheiling@live.unc.edu>

**Description** Fits penalized high and low dimensional generalized linear mixed models using the Monte Carlo Expectation Conditional Minimization (MCECM) algorithm and coordinate descent.
Supports fitting of logistic regression models and MCP penalties.

**License** GPL (>= 2)

**Encoding** UTF-8

**Imports** ggplot2,
Matrix,
methods,
ncvreg,
reshape2,
rstan (>= 2.18.1),
rstantools (>= 2.0.0),
stringr,
mvtnorm,
MASS

**Depends** lme4,
BH (>= 1.66.0),
bigmemory,
R (>= 3.6.0),
Rcpp (>= 0.12.0),
RcppArmadillo,
RcppEigen (>= 0.3.3.3.0)

**LinkingTo** BH (>= 1.66.0),
bigmemory,
Rcpp (>= 0.12.0),
RcppArmadillo,
RcppDist,
RcppEigen (>= 0.3.3.3.0),
rstan (>= 2.18.1),
StanHeaders (>= 2.18.0)

**RoxygenNote** 7.1.1

**NeedsCompilation** yes

**Author** Hillary Heiling [aut, cre],
Naim Rashid [aut],
Quefeng Li [aut]

**Suggests** testthat,
knitr,
rmarkdown

**Biarch** true

**SystemRequirements** GNU make

**VignetteBuilder** knitr

# R topics documented:



---

`adaptControl` *Control of Metropolis-within-Gibbs Adaptive Random Walk Sampling Procedure Controls the adaptive random walk Metropolis-within-Gibbs sampling procedure.*

---

## Description

Control of Metropolis-within-Gibbs Adaptive Random Walk Sampling Procedure

Controls the adaptive random walk Metropolis-within-Gibbs sampling procedure.

## Usage

```
adaptControl(batch_length = 100, offset = 0)
```

### Arguments

| | |
|---|---|
| batch_length | positive integer specifying the number of posterior draws to collect before the proposal variance is adjusted based on the acceptance rate of the last batch_length accepted posterior draws. Default is set to 100. Batch length restricted to be no less than 50. |
| offset | non-negative integer specifying an offset value for the increment of the proposal variance adjustment. Optionally used to ensure the required diminishing adaptation condition. Default set to 0.<br>increment = min(0.01, 1 / sqrt(batch*batch_length + offset)) |

### Value

Function returns a list (inheriting from class "adaptControl") containing parameter specifications for the adaptive random walk sampling procedure.

---

## fit_dat_B *Fit a Penalized Generalized Mixed Model via Monte Carlo Expectation Conditional Minimization (MCECM)*

---

### Description

Description

### Usage

```
fit_dat_B(
  dat,
  lambda0 = 0,
  lambda1 = 0,
  conv_EM = 0.001,
  conv_CD = 1e-04,
  family = "binomial",
  offset_fit = NULL,
  trace = 0,
  penalty = c("MCP", "SCAD", "lasso"),
  alpha = 1,
  gamma_penalty = switch(penalty[1], SCAD = 4, 3),
  group_X = 0:(ncol(dat$X) - 1),
  nMC_burnin = 250,
  nMC = 250,
  nMC_max = 5000,
  t = 2,
  mcc = 2,
  u_init = NULL,
  coef_old = NULL,
  ufull_describe = NULL,
  maxitEM = 50,
  maxit_CD = 250,
  M = 10^4,
  sampler = c("stan", "random_walk", "independence"),
```

```
  adapt_RW_options = adaptControl(),
  covar = c("unstructured", "independent"),
  var_start = 1,
  logLik_calc = F,
  max_cores = 1,
  checks_complete = F,
  ranef_keep = rep(1, times = (ncol(dat$Z)/nlevels(dat$group)))
)
```

## Arguments

| | |
|---|---|
| dat | a list object specifying y (response vector), X (model matrix of all covariates), Z (model matrix for the random effects), and group (numeric factor vector whose value indicates the study, batch, or other group identity to which on observation belongs) |
| lambda0 | a non-negative numeric penalty parameter for the fixed effects parameters |
| lambda1 | a non-negative numeric penalty parameter for the (grouped) random effects covariance parameters |
| conv_EM | a non-negative numeric convergence criteria for the convergence of the EM algorithm. Default is 0.001. EM algorithm is considered to have converge if the average Euclidean distance between the current coefficient estimates and the coefficient estimates from t EM iterations back is less than conv_EM mcc times in a row. See t and mcc for more details. |
| conv_CD | a non-negative numeric convergence criteria for the convergence of the grouped coordinate descent loop within the M step of the EM algorithm. Default 0.0001. |
| family | a description of the error distribution and link function to be used in the model. Currently, the glmmPen algorithm allows the binomial, gaussian, and poisson families with canonical links only. |
| offset_fit | This can be used to specify an a priori known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases. |
| trace | an integer specifying print output to include as function runs. Default value is 0. See Details for more information about output provided when trace = 0, 1, or 2. |
| penalty | character describing the type of penalty to use in the variable selection procedure. Options include 'MCP', 'SCAD', and 'lasso'. Default is MCP penalty. If the random effect covariance matrix is "unstructured", then a group MCP, group SCAD, or group Lasso penalty is used on the random effects coefficients. |
| alpha | Tuning parameter for the Mnet estimator which controls the relative contributions from the MCP/SCAD/lasso penalty and the ridge, or L2, penalty. alpha=1 is equivalent to the MCP/SCAD/lasso penalty, while alpha=0 is equivalent to ridge regression. However, alpha=0 is not supported; alpha may be arbibrarily small, but not exactly zero |
| gamma_penalty | The tuning parameter of the MCP and SCAD penalties. Not used by Lasso penalty. Default is 4.0 for SCAD and 3.0 for MCP. |
| group_X | vector describing the grouping of the covariates in the model matrix. |
| nMC_burnin | positive integer specifying the number of posterior draws to use as burnin for each E step in the EM algorithm. If set to NULL, the algorithm inputs the following defaults: Default 250 when the number of random effects predictors is less than or equal to 10; default 100 otherwise. Function will not allow nMC_burnin to be less than 100. |

nMC — a positive integer for the initial number of Monte Carlo draws. See the nMC_start argument in optimControl for more details.

nMC_max — a positive integer for the maximum number of allowed Monte Carlo draws used in each step of the EM algorithm. If set to NULL, the algorithm inputs the following defaults: When the number of random effect predictors is 10 or less, Default is set to 5000 when no selection is performed and 2500 when selection is performed. Default is set to 1000 when the number of random effect predictors is greater than 10.

t — the convergence criteria is based on the average Euclidean distance between the most recent coefficient estimates and the coefficient estimates from t EM iterations back. Positive integer, default equals 2.

mcc — the number of times the convergence critera must be met before the algorithm is seen as having converged (mcc for 'meet condition counter'). Default set to 2. Value retricted to be no less than 2.

maxitEM — a positive integer for the maximum number of allowed EM iterations. If set to NULL, then the algorithm inputs the following defaults: Default equals 50 for the Binomial and Poisson families, 100 for the Gaussian family.

maxit_CD — a positive integer for the maximum number of allowed interations for the coordinate descent algorithms used within the M-step of each EM iteration. Default equals 50.

M — positive integer specifying the number of posterior draws to use within the Pajor log-likelihood calculation. Default is 10^4; minimum allowed value is 5000.

sampler — character string specifying whether the posterior draws of the random effects should be drawn using Stan (default, from package rstan) or the Metropolis-within-Gibbs procedure incorporating an adaptive random walk sampler ("random_walk") or an independence sampler ("independence"). If using the random walk sampler, see adaptControl for some additional control structure parameters.

adapt_RW_options — a list of class "adaptControl" from function adaptControl containing the control parameters for the adaptive random walk Metropolis-within-Gibbs procedure. Ignored if optimControl parameter sampler is set to "stan" (default) or "independence".

covar — character string specifying whether the covariance matrix should be unstructured ("unstructured") or diagonal with no covariances between variables ("independent"). Default is set to NULL. If covar is set to NULL and the number of random effects predictors (not including the intercept) is greater than or equal to 10 (i.e. high dimensional), then the algorithm automatically assumes an independent covariance structure and covar is set to "independent". Otherwise if covar is set to NULL and the number of random effects predictors is less than 10, then the algorithm automatically assumes an unstructured covariance structure and covar is set to "unstructured".

var_start — either the character string "recommend" or a positive number specifying the starting values to initialize the variance of the covariance matrix. Default "recommend" first fits a simple model with a fixed and random intercept only using a Laplace estimate. The random intercept variance estimate from this model is then multiplied by 2 and used as the starting variance.

logLik_calc — logical value specifying if the log likelihood (and log-likelihood based calculations BIC, BICh, and BICNgrp) should be calculated for all of the models in

the selection procedure. If BIC-ICQ is used for selection, the log-likelihood is not needed. However, if users are interested in comparing the best models from BIC-ICQ and other BIC-type selection criteria, setting `logLik_calc` to `TRUE` will calculate these other options for all of the models.

`max_cores` integer describing the number of cores available for computation. If `max_cores` is specified to be greater than 1 and the sampler is specified as "stan", then parallel computation using multiple cores is used to calculate the Stan MCMC samples within each E step. The package authors do not advise using parallelization in the E step unless `nMC_max` is set to a large number, such as 10^4 or more.

`checks_complete`
boolean value indicating whether the function has been called within `glmm` or `glmmPen` or whether the function has been called by itself. If true, performs additional checks on the input data. If false, assumes data input checks have already been performed. For package testing purposes only.

## Value

a list with the following elements:

`coef` a numeric vector of coefficients of fixed effects estimates and non-zero estimates of the lower-triangular cholesky decomposition of the random effects covariance matrix (in vector form)

`sigma` random effects covariance matrix

`lambda0, lambda1`
the penalty parameters input into the function

`covgroup` Organization of how random effects coefficients are grouped.

`J` a sparse matrix that transforms the non-zero elements of the lower-triangular cholesky decomposition of the random effects covariance matrix into a vector. For unstructured covariance matrices, dimension of dimension q^2 x (q(q+1)/2) (where q = number of random effects). For independent covariance matrices, q^2 x q.

`ll` estimate of the log likelihood, calculated using the Pajor method

`BICh` the hybrid BIC estimate described in Delattre, Lavielle, and Poursat (2014)

`BIC` Regular BIC estimate

`BICq` BIC-ICQ estimate

`u` a matrix of the Monte Carlo draws. Organization of columns: first by random effect variable, then by group within variable (i.e. Var1:Grp1 Var1:Grp2 ... Var1:GrpK Var2:Grp1 ... Varq:GrpK)

`gibbs_accept_rate`
a matrix of the ending gibbs acceptance rates for each variable (columns) and each group (rows) when the sampler is either "random_walk" or "independence"

`proposal_SD` a matrix of the ending proposal standard deviations (used in the adaptive random walk version of the Metropolis-within-Gibbs sampling) for each variable (columns) and each group (rows)

`glFormula_edit` *Extracting Useful Vectors and Matrices from Formula and Data Information*

## Description

Takes the model `formula` and an optional data frame and converts them into y, X, Z, and group output.

## Usage

```
glFormula_edit(
  formula,
  data = NULL,
  family,
  subset,
  weights,
  na.action,
  offset,
  contrasts = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| `formula` | a two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a ~ operator and the terms, sepearated by + operators, on the right. Random-effects terms are distinguished by vertical bars ("\|") separating expression for design matrices from grouping factors. `formula` should be of the same format needed for `glmer` in package **lme4**. Only one grouping factor will be recognized. The random effects covariates need to be a subset of the fixed effects covariates. The offset must be specified outside of the formula in the 'offset' argument. |
| `data` | an optional data frame containing the variables named in `formula`. Although `data` is optional, the package authors *strongly* recommend its use. If `data` is omitted, variables will be taken from the environment of `formula` (if specified as a formula). |
| `na.action` | a function that indicates what should happen when the data contain NAs. The default option `na.omit` removes observations with any missing values in any of the variables |

## Value

a list with the following elements:

| | |
|---|---|
| `y` | a numeric vector of the response variable |
| `X` | a model matrix with the fixed covariates |
| `Z` | a sparse model matrix for the random effects |
| `group` | a factor vector of the grouping variable |
| `cnms` | a vector of column names of the random effects |

| | |
|---|---|
| group_name | character name of the group variable |
| flist | a list of grouping factors using inf the random-effects terms |
| frame | a model frame including all fixed and random covariates, the response, and the grouping variable |

---

glmm *Fit a Generalized Mixed Model via Monte Carlo Expectation Conditional Minimization (MCECM)*

---

## Description

glmm is used to fit a single generalized mixed model via Monte Carlo Expectation Conditional Minimization (MCECM). Unlike glmmPen, no model selection is performed.

## Usage

```
glmm(
  formula,
  data = NULL,
  family = "binomial",
  covar = c("unstructured", "independent"),
  offset = NULL,
  na.action = na.omit,
  optim_options = optimControl(),
  adapt_RW_options = adaptControl(),
  trace = 0,
  tuning_options = lambdaControl(),
  ...
)
```

## Arguments

| | |
|---|---|
| formula | a two-sided linear formula object describing both the fixed effects and random effects part of the model, with the response on the left of a ~ operator and the terms, sepearated by + operators, on the right. Random-effects terms are distinguished by vertical bars ("\|") separating expression for design matrices from the grouping factor. formula should be of the same format needed for glmer in package **lme4**. Only one grouping factor will be recognized. The random effects covariates need to be a subset of the fixed effects covariates. The offset must be specified outside of the formula in the 'offset' argument. |
| data | an optional data frame containing the variables named in formula. If data is omitted, variables will be taken from the environment of formula (if specified as a formula). |
| family | a description of the error distribution and link function to be used in the model. Currently, the glmmPen algorithm allows the binomial, gaussian, and poisson families with canonical links only. |
| covar | character string specifying whether the covariance matrix should be unstructured ("unstructured") or diagonal with no covariances between variables ("independent"). Default is set to NULL. If covar is set to NULL and the number of random effects predictors (not including the intercept) is greater than or equal to 10 (i.e. |

high dimensional), then the algorithm automatically assumes an independent covariance structure and covar is set to "independent". Otherwise if covar is set to NULL and the number of random effects predictors is less than 10, then the algorithm automatically assumes an unstructured covariance structure and covar is set to "unstructured".

offset — This can be used to specify an *a priori* known component to be included in the linear predictor during fitting. Default set to NULL (no offset). If the data argument is NULL, this should be a numeric vector of length equal to the number of cases (the response). If the data argument specifies a data.frame, the offset argument should specify the name of a column in the data.frame.

na.action — a function that indicates what should happen when the data contain NAs. Only the option na.omit are recognized by this function.

optim_options — a structure of class "optimControl" created from function optimControl that specifies optimization parameters. See the documentation for optimControl for more details on defaults.

adapt_RW_options — a list of class "adaptControl" from function adaptControl containing the control parameters for the adaptive random walk Metropolis-within-Gibbs procedure. Ignored if optimControl parameter sampler is set to "stan" (default) or "independence".

trace — an integer specifying print output to include as function runs. Default value is 0. See Details for more information about output provided when trace = 0, 1, or 2.

tuning_options — a list of class selectControl or lambdaControl resulting from selectControl or lambdaControl containing additional control parameters. When function glmm is used,the algorithm may be run using one specific set of penalty parameters lambda0 and lambda1 by specifying such values in lambdaControl(). The default for glmm is to run the model fit with no penalization (lambda0 = lambda1 = 0). When function glmmPen is run, tuning_options is specified usig selectControl{}. See the lambdaControl and selectControl documentation for further details.

... — additional arguments that could be passed into glmmPen. See glmmPen for further details.

## Details

The glmm function can be used to fit a single generalized mixed model. While this approach is meant to be used in the 'oracle' case where the user knows which covariates belong in the fixed and random effects and no penalization is required, one is allowed to specify non-zero fixed and random effects penalites using lambdaControl and the (...) arguments. The (...) allow for specification of penalty-related arguments; see glmmPen for details. For a high dimensional situation, the user may want to fit a full model using a small penalty for the fixed and random effects and save the posterior draws from this full model for use in any BIC-ICQ calculations during selection within glmmPen. Specifying a .txt file in the 'BICq_posterior' argument will save the posterior draws from the glmm model into a big.matrix within the .txt file specified (bigmemory::write.big.matrix).

## Value

A reference class object of class pglmmObj for which many methods are available (e.g. methods(class = "pglmmObj"))

glmmPen *Fit Penalized Generalized Mixed Models via Monte Carlo Expectation Conditional Minimization (MCECM)*

## Description

glmmPen is used to fit penalized generalized mixed models via Monte Carlo Expectation Conditional Minimization (MCECM) and select the best model using BIC-type selection criteria

## Usage

```
glmmPen(
  formula,
  data = NULL,
  family = "binomial",
  covar = NULL,
  offset = NULL,
  na.action = na.omit,
  fixef_noPen = NULL,
  penalty = c("MCP", "SCAD", "lasso"),
  alpha = 1,
  gamma_penalty = switch(penalty[1], SCAD = 4, 3),
  optim_options = optimControl(),
  adapt_RW_options = adaptControl(),
  trace = 0,
  tuning_options = selectControl(),
  BICq_posterior = NULL
)
```

## Arguments

| | |
|---|---|
| formula | a two-sided linear formula object describing both the fixed effects and random effects part of the model, with the response on the left of a ~ operator and the terms, sepearated by + operators, on the right. Random-effects terms are distinguished by vertical bars ("\|") separating expression for design matrices from the grouping factor. formula should be of the same format needed for glmer in package **lme4**. Only one grouping factor will be recognized. The random effects covariates need to be a subset of the fixed effects covariates. The offset must be specified outside of the formula in the 'offset' argument. |
| data | an optional data frame containing the variables named in formula. If data is omitted, variables will be taken from the environment of formula (if specified as a formula). |
| family | a description of the error distribution and link function to be used in the model. Currently, the glmmPen algorithm allows the binomial, gaussian, and poisson families with canonical links only. |
| covar | character string specifying whether the covariance matrix should be unstructured ("unstructured") or diagonal with no covariances between variables ("independent"). Default is set to NULL. If covar is set to NULL and the number of random effects predictors (not including the intercept) is greater than or equal to 10 (i.e. high dimensional), then the algorithm automatically assumes an independent |

covariance structure and covar is set to "independent". Otherwise if covar is set to NULL and the number of random effects predictors is less than 10, then the algorithm automatically assumes an unstructured covariance structure and covar is set to "unstructured".

offset — This can be used to specify an *a priori* known component to be included in the linear predictor during fitting. Default set to NULL (no offset). If the data argument is NULL, this should be a numeric vector of length equal to the number of cases (the response). If the data argument specifies a data.frame, the offset argument should specify the name of a column in the data.frame.

na.action — a function that indicates what should happen when the data contain NAs. Only the option na.omit are recognized by this function.

fixef_noPen — Optional vector of 0's and 1's of the same length as the number of fixed effects covariates used in the model. Value 0 indicates the variable should not have its fixed effect coefficient penalized, 1 indicates that it can be penalized. Order should correspond to the same order of the fixed effects given in the formula.

penalty — character describing the type of penalty to use in the variable selection procedure. Options include 'MCP', 'SCAD', and 'lasso'. Default is MCP penalty. If the random effect covariance matrix is "unstructured", then a group MCP, group SCAD, or group Lasso penalty is used on the random effects coefficients.

alpha — Tuning parameter for the Mnet estimator which controls the relative contributions from the MCP/SCAD/lasso penalty and the ridge, or L2, penalty. alpha=1 is equivalent to the MCP/SCAD/lasso penalty, while alpha=0 is equivalent to ridge regression. However, alpha=0 is not supported; alpha may be arbibrarily small, but not exactly zero

gamma_penalty — The tuning parameter of the MCP and SCAD penalties. Not used by Lasso penalty. Default is 4.0 for SCAD and 3.0 for MCP.

optim_options — a structure of class "optimControl" created from function optimControl that specifies optimization parameters. See the documentation for optimControl for more details on defaults.

adapt_RW_options — a list of class "adaptControl" from function adaptControl containing the control parameters for the adaptive random walk Metropolis-within-Gibbs procedure. Ignored if optimControl parameter sampler is set to "stan" (default) or "independence".

trace — an integer specifying print output to include as function runs. Default value is 0. See Details for more information about output provided when trace = 0, 1, or 2.

tuning_options — a list of class selectControl or lambdaControl resulting from selectControl or lambdaControl containing additional control parameters. When function glmm is used,the algorithm may be run using one specific set of penalty parameters lambda0 and lambda1 by specifying such values in lambdaControl(). The default for glmm is to run the model fit with no penalization (lambda0 = lambda1 = 0). When function glmmPen is run, tuning_options is specified usig selectControl{}. See the lambdaControl and selectControl documentation for further details.

BICq_posterior — an optional character string expressing the path and file basename of a file combination that the will file-back or currently file-backs a big.matrix of the posterior draws from the full model. These full model posterior draws will be used in BIC-ICQ calculations if these calculations are requested. If this argument is specified as NULL (default) and BIC-ICQ calculations are requested, the posterior draws will be saved in the file combination 'BICq_Posterior_Draws.bin'

and 'BICq_Posterior_Draws.desc' in the working directory. See 'Details' section for additional details about the required format of `BICq_posterior` and the file-backed big matrix.

## Value

A reference class object of class `pglmmObj` for which many methods are available (e.g. `methods(class = "pglmmObj")`)

---

`glmmPen_FineSearch` *Fit a Penalized Generalized Mixed Model via Monte Carlo Expectation Conditional Minimization (MCECM) using a finer penalty grid search* `glmmPen_FineSearch` *finds the best model from the selection results of a* `pglmmObj` *object created by* `glmmPen`*, identifies a more targeted grid search around the optimum lambda penalty values, and performs model selection on this finer grid search.*

---

## Description

Fit a Penalized Generalized Mixed Model via Monte Carlo Expectation Conditional Minimization (MCECM) using a finer penalty grid search

`glmmPen_FineSearch` finds the best model from the selection results of a `pglmmObj` object created by `glmmPen`, identifies a more targeted grid search around the optimum lambda penalty values, and performs model selection on this finer grid search.

## Usage

```
glmmPen_FineSearch(
  object,
  tuning_options = selectControl(),
  idx_range = 2,
  optim_options = NULL,
  adapt_RW_options = NULL,
  trace = 0,
  BICq_posterior = NULL
)
```

## Arguments

| | |
|---|---|
| `object` | an object of class `pglmmObj` created by `glmmPen`. This object must contain model selection results. |
| `tuning_options` | a list of class selectControl resulting from `selectControl` containing model selection control parameters. See the `selectControl` documentation for details. The user can specify their own fine grid search, or if the lambda0_seq and lambda1_seq arguments are left as `NULL`, the algorithm will automatically select a fine grid search based on the best model from the previous selection. See Details for more information. Default value set to 1. |
| `idx_range` | a positive integer that determines what positions within the sequence of the fixed and random effect lambda penalty parameters used in the previous coarse grid search will be used as the new fixed and random effect lambda penalty parameter ranges. See Details for more information. |

`optim_options` an optional list of class "optimControl" created from function `optimControl` that specifies optimization parameters. If set to the default `NULL`, will use the optimization parameters used for the previous round of selection stored within the `pglmmObj` object.

`adapt_RW_options` an optional list of class "adaptControl" from function `adaptControl` containing the control parameters for the adaptive random walk Metropolis-within-Gibbs procedure. Ignored if `optimControl` parameter `sampler` is set to "stan" or "independence". If set to the default `NULL`, will use the adaptive random walk paraters used for the previous round of selection stored within the `pglmmObj` object.

`trace` an integer specifying print output to include as function runs. Default value is 0. See Details for more information about output provided when trace = 0, 1, or 2.

`BICq_posterior` an optional character string specifying the file-backed `big.matrix` containing the posterior draws used to calculate the BIC-ICQ selection criterion if such a `big.matrix` was created in the previous round of selection. See `glmmPen` documentation for further details.

## Details

The `glmmPen_FineSearch` function extracts the data, the penalty information (penalty type, gamma_penalty, and alpha), and some other argument specifications from the `pglmmObj` object created during a previous round of model selection. In this finer grid search, the user has the ability to make the following adjustments: the user can change the BIC option used for selection, any optimization control parameters, or any adaptive random walk parameters (if the sampler specified in the optimization parameters is "random_walk"). The user could manually specify the lambda penalty grid to search over within the `selectControl` control parameters, or the user could let the `glmmPen_FineSearch` algorithm calculate a finer grid search automatically (see next paragraph for details).

If the sequences of lambda penalty values are left unspecified in the `selectControl` tuning options, the `glmmPen_FineSearch` algorithm performs the following steps to find the finer lambda grid search: (i) The lambda combination from the best model is identified from the earlier selection results saved in the `pglmmObj` object. (ii) For the fixed and random effects separately, the new max and min lambda values are the lambda values `idx_range` positions away from the best lambda in the original lambda sequences for the fixed and random effects.

Trace details: The value of 0 outputs some general updates for each EM iteration (iteration number EM_iter, number of MCMC draws nMC, average Euclidean distance between current coefficients and coefficients from t iterations back EM_diff, and number of non-zero coefficients Non0 Coef). The value of 1 additionally outputs the updated coefficients, updated covariance matrix values, and the number of coordinate descent iterations used for the M step for each EM iteration. The value of 2 outputs all of the above plus gibbs acceptance rate information for the adaptive random walk and independence samplers and the updated proposal standard deviation for the adaptive random walk.

## Value

A reference class object of class `pglmmObj` for which many methods are available (e.g. `methods(class = "pglmmObj")`)

`lambdaControl` *Control of Penalization Parameters and Selection Criteria*

## Description

Constructs control structures for penalized mixed model fitting.

## Usage

```
lambdaControl(lambda0 = 0, lambda1 = 0)

selectControl(
  lambda0_seq = NULL,
  lambda1_seq = NULL,
  nlambda = 10,
  search = c("abbrev", "full_grid"),
  BIC_option = c("BICq", "BICh", "BIC", "BICNgrp"),
  logLik_calc = switch(BIC_option[1], BICq = F, T),
  lambda.min = NULL,
  pre_screen = T,
  lambda.min.presc = NULL
)
```

## Arguments

| | |
|---|---|
| `lambda0` | a non-negative numeric penalty parameter for the fixed effects parameters |
| `lambda1` | a non-negative numeric penalty parameter for the (grouped) random effects covariance parameters |
| `lambda0_seq, lambda1_seq` | a sequence of non-negative numeric penalty parameters for the fixed and random effect parameters, respectively. If `NULL`, then a sequence will be automatically calculated. See 'Details' section for more details on these default calculations. |
| `nlambda` | positive integer specifying number of penalty parameters (lambda) with which to fit a model. Ignored if `lambda0_seq` and `lambda1_seq` are specified by the user. |
| `search` | character string of "abbrev" (default) or "full_grid" indicating if the search of models over the penalty parameter space should be the full grid search (total number of models equals 'nlambda'^2 or length('lambda0_seq')*length('lambda1_seq')) or an abbreviated grid search. The abbreviated grid search is described in more detail in the Details section. |
| `BIC_option` | character string specifing the selection criteria used to select the 'best' model. Default "BICq" option specifies the BIC-ICQ criterion, which requires a fit of a full model; a small penalty of 0.001*(lambda max) is used for the fixed and/or random effects if the number of effects is 10 or greater. The "BICh" option utilizes the hybrid BIC value described in Delattre, Lavielle, and Poursat (2014). The regular "BIC" option penalty term uses (total non-zero coefficients)*(length(y) = total number observations). The "BICNgrp" option penalty term uses (total non-zero coefficients)*(nlevels(group) = number groups). |

`logLik_calc` logical value specifying if the log likelihood (and log-likelihood based calculations BIC, BICh, and BICNgrp) should be calculated for all of the models in the selection procedure. If BIC-ICQ is used for selection, the log-likelihood is not needed. However, if users are interested in comparing the best models from BIC-ICQ and other BIC-type selection criteria, setting `logLik_calc` to `TRUE` will calculate these other options for all of the models.

`lambda.min` numeric fraction between 0 and 1. The sequence of the lambda penalty parameters ranges from the maximum lambda where all fixed and random effects are penalized to 0 and a minimum lambda value, which equals a small fraction of the maximum lambda. The parameter `lambda.min` specifiex this fraction. Default value is set to `NULL`, which automatically selects `lambda.min` to equal 0.01 when n < p and 0.05 when p >= n.

`pre_screen` logical value indicating whether pre-screening should be performed before model selection (default `TRUE`). If the number of random effects considered less than 5, no pre-screening will be performed. Pre-screening removes random effects from consideration during the model selection process, which can significantly speed up the algorithm.

`lambda.min.presc`
numeric fraction between 0 and 1. During pre-screening and the full model fit for the BIC-ICQ calculation, the small penalty used on the random effect is the fraction `lambda.min.presc` mulitplied by the maximum penalty parameter that penalizes all fixed and random effects to 0. If left as `NULL`, the default value is 0.01 when the number of random effects is 10 or less and 0.05 otherwise.

## Details

If unspecified, the `lambda0_seq` and `lambda1_seq` numeric sequences are automatically calculated. The sequence will be calculated in the same manner as `ncvreg` calculates the range: max penalizes all fixed and random effects to 0, min is a small portion of max (`lambda.min`*(lambda max)), sequence is composed of `nlambda` values spread evenly on the log scale. Unlike `ncvreg`, the order of penalty values used in the algorithm must run from the min lambda to the max lambda (as opposed to running from max lambda to min lambda). The length of the sequence is specified by `nlambda`. By default, these sequences are calculated using `LambdaSeq`.

For secondary rounds run using `glmmPen_FineSearch`, the optimal lambda combination from the initial round (based on the specified `BIC_option`) will be used to calculate a finer grid search. The new max and min lambda values are the lambda values `idx_lambda` positions away from the best lambda in the original lambda sequences for the fixed and random effects.

The `lambda0` and `lambda1` arguments allow for a user to fit a model with a single non-zero penalty parameter combination. However, this is generally not recommended.

Abbreviated grid search: The abbreviated grid search proceeds in two stages. In stage 1, the algorithm fits the following series of models: the fixed effects penalty parameter remains a fixed value evaluated at the minimum of the fixed effects penalty parameters, and a all random effects penalty parameters are examined. The 'best' model from this first stage of models determines the optimum random effect penalty parameter. In stage 2, the algorithm fits the following series of models: the random effects penalty parameter remains fixed at the value of the optimum random effect penalty parameter (from stage 1) and all fixed effects penalty parameters are considered. The best overall model is the best model from stage 2. This reduces the number of models considered to length(‘lambda0_seq‘) + length(‘lambda1_seq‘). Although this can drastically increase the time of the algorithm to proceed, the authors found in simulations that this generally increases the false positive rate by a factor of between 1.5 and 2.

## Value

The *Control functions return a list (inheriting from class "pglmmControl") containing parameter values that determine settings for variable selection.

---

LambdaSeq *Calculation of Penalty Parameter Sequence (Lambda Sequence)*

---

## Description

Calculates the sequence of penalty parameters used in the model selection procedure. This function calls functions from package ncvreg.

## Usage

```
LambdaSeq(
  X,
  y,
  family,
  alpha = 1,
  lambda.min = NULL,
  nlambda = 10,
  penalty.factor = NULL
)
```

## Arguments

| | |
|---|---|
| X | matrix of standardized fixed effects (see std function in ncvreg documenation). X should not include intercept. |
| y | numeric vector of response values |
| family | a description of the error distribution and link function to be used in the model. Currently, the glmmPen algorithm allows the binomial, gaussian, and poisson families with canonical links only. |
| alpha | Tuning parameter for the Mnet estimator which controls the relative contributions from the MCP/SCAD/lasso penalty and the ridge, or L2, penalty. alpha=1 is equivalent to the MCP/SCAD/lasso penalty, while alpha=0 is equivalent to ridge regression. However, alpha=0 is not supported; alpha may be arbibrarily small, but not exactly zero |
| lambda.min | numeric fraction between 0 and 1. The sequence of the lambda penalty parameters ranges from the maximum lambda where all fixed and random effects are penalized to 0 and a minimum lambda value, which equals a small fraction of the maximum lambda. The parameter lambda.min specifiex this fraction. Default value is set to NULL, which automatically selects lambda.min to equal 0.01 when n < p and 0.05 when p >= n. |
| nlambda | positive integer specifying number of penalty parameters (lambda) with which to fit a model. |
| penalty.factor | an optional numeric vector equal to the fixef_noPen argument in glmmPen |

## Value

numeric sequence of penalty parameters of length `nlambda` ranging from the minimum penalty parameter (first element) equal to fraction `lambda.min` multiplied by the maximum penalty parameter to the maximum penalty parameter (last element)

---

`optimControl` *Control of Penalized Generalized Linear Mixed Model Fitting Constructs the control structure for the optimization of the penalized mixed model fit algorithm.*

---

## Description

Control of Penalized Generalized Linear Mixed Model Fitting

Constructs the control structure for the optimization of the penalized mixed model fit algorithm.

## Usage

```
optimControl(
  conv_EM = 0.0015,
  conv_CD = 5e-04,
  nMC_burnin = NULL,
  nMC_start = NULL,
  nMC_max = NULL,
  nMC_report = 5000,
  maxitEM = NULL,
  maxit_CD = 50,
  M = 10000,
  t = 2,
  mcc = 2,
  sampler = c("stan", "random_walk", "independence"),
  var_start = "recommend",
  max_cores = 1
)
```

## Arguments

| | |
|---|---|
| `conv_EM` | a non-negative numeric convergence criteria for the convergence of the EM algorithm. Default is 0.001. EM algorithm is considered to have converge if the average Euclidean distance between the current coefficient estimates and the coefficient estimates from `t` EM iterations back is less than `conv_EM` `mcc` times in a row. See `t` and `mcc` for more details. |
| `conv_CD` | a non-negative numeric convergence criteria for the convergence of the grouped coordinate descent loop within the M step of the EM algorithm. Default 0.0001. |
| `nMC_burnin` | positive integer specifying the number of posterior draws to use as burnin for each E step in the EM algorithm. If set to `NULL`, the algorithm inputs the following defaults: Default 250 when the number of random effects predictors is less than or equal to 10; default 100 otherwise. Function will not allow `nMC_burnin` to be less than 100. |

`nMC_start` a positive integer for the initial number of Monte Carlo draws. If set to `NULL`, the algorithm inputs the following defaults: Default 250 when the number of random effects predictors is less than or equal to 10; default 100 otherwise.

`nMC_max` a positive integer for the maximum number of allowed Monte Carlo draws used in each step of the EM algorithm. If set to `NULL`, the algorithm inputs the following defaults: When the number of random effect predictors is 10 or less, Default is set to 5000 when no selection is performed and 2500 when selection is performed. Default is set to 1000 when the number of random effect predictors is greater than 10.

`nMC_report` a positive integer for the number of posterior draws to save from the final model. These posterior draws can be used for diagnostic purposes, see `plot_mcmc`

`maxitEM` a positive integer for the maximum number of allowed EM iterations. If set to `NULL`, then the algorithm inputs the following defaults: Default equals 50 for the Binomial and Poisson families, 100 for the Gaussian family.

`maxit_CD` a positive integer for the maximum number of allowed interations for the coordinate descent algorithms used within the M-step of each EM iteration. Default equals 50.

`M` positive integer specifying the number of posterior draws to use within the Pajor log-likelihood calculation. Default is 10^4; minimum allowed value is 5000.

`t` the convergence criteria is based on the average Euclidean distance between the most recent coefficient estimates and the coefficient estimates from t EM iterations back. Positive integer, default equals 2.

`mcc` the number of times the convergence critera must be met before the algorithm is seen as having converged (mcc for 'meet condition counter'). Default set to 2. Value retricted to be no less than 2.

`sampler` character string specifying whether the posterior draws of the random effects should be drawn using Stan (default, from package rstan) or the Metropolis-within-Gibbs procedure incorporating an adaptive random walk sampler ("random_walk") or an independence sampler ("independence"). If using the random walk sampler, see `adaptControl` for some additional control structure parameters.

`var_start` either the character string "recommend" or a positive number specifying the starting values to initialize the variance of the covariance matrix. Default "recommend" first fits a simple model with a fixed and random intercept only using a Laplace estimate. The random intercept variance estimate from this model is then multiplied by 2 and used as the starting variance.

`max_cores` integer describing the number of cores available for computation. If `max_cores` is specified to be greater than 1 and the sampler is specified as "stan", then parallel computation using multiple cores is used to calculate the Stan MCMC samples within each E step. The package authors do not advise using parallelization in the E step unless `nMC_max` is set to a large number, such as 10^4 or more.

## Details

When the `optim_options` arugment in `glmm` and `glmmPen` is set to "recommend", the default settings discussed in the given `optimControl` arguments are used. These default settings depend on both the family of the data structure and the number of random effects predictors specified for use. If the user specifies `optim_options = optimControl()` with any argument specifications, no additional adjustments will be performed on the arguments based on family or random effect predictors.

## Value

Function returns a list (inheriting from class "optimControl") containing fit and optimization criteria values used in optimization routine.

---

`pglmmObj-class` *Class "pglmmObj" of Fitted Penalized Generalized Mixed-Effects Models*

---

## Description

The functions `glmm`, `glmmPen`, and `glmmPen_FineSearch` output the reference class object of type `pglmmObj`.

## Examples

```
showClass("pglmmObj")
methods(class = "pglmmObj")
```

---

`plot_mcmc` *Plot Diagnostics for MCMC Posterior Draws of the Random Effects*

---

## Description

Provides graphical diagnostics of the random effect posterior draws from the (best) model. Availabile diagnostics include the sample path, histograms, cummulative sums, and autocorrelation.

## Usage

```
plot_mcmc(
  object,
  plots = c("all", "sample.path", "histogram", "cumsum", "autocorr"),
  grps = "all",
  vars = "all",
  numeric.grps = F,
  bin_width = NULL
)
```

## Arguments

| | |
|---|---|
| `object` | an object of class `pglmmObj` output from either `glmmPen` or `glmmPen_FineSearch`. |
| `plots` | a character string or a vector of character strings specifying which graphical diagnostics to provide. |
| `grps` | a character string or a vector of character strings specifying which groups should have diagnostics provided. The names of the groups match the input group factor levels. Default is set to 'all' for all groups. |

| | |
|---|---|
| vars | a character string or a vector of character strings specifying which variables should have diagnostics provided. Tip: can find the names of the random effect variables in the output sigma matrix found in the pglmmObj object. Default is set to 'all', which picks all variables with non-zero random effects. |
| numeric.grps | if TRUE, specifies that the groups factor should be converted to numeric values. This option could be used to ensure that the organization of the groups is in the proper numeric order. |
| bin_width | optional binwidth argument for geom_histogram from the ggplot2 package. Default set to NULL, which specifies the default geom_histogram binwidth. |

## Value

a list of ggplot graphics, each faceted by group and random effect variable. Type of plots specified in the plots argument.

---

# select_tune *Fit a Penalized Generalized Mixed Model via Monte Carlo Expectation Conditional Minimization (MCECM)*

---

## Description

select_tune is used to fit penalized generalized mixed models via Monte Carlo Expectation Conditional Minimization (MCECM) over multiple tuning parameters and is called within glmmPen

## Usage

```
select_tune(
  dat,
  offset = NULL,
  family,
  covar = c("unstructured", "independent"),
  group_X = 0:(ncol(dat$X) - 1),
  penalty,
  lambda0_seq,
  lambda1_seq,
  alpha = 1,
  gamma_penalty = switch(penalty[1], SCAD = 4, 3),
  trace = 0,
  u_init = NULL,
  coef_old = NULL,
  adapt_RW_options = adaptControl(),
  optim_options = optimControl(),
  BIC_option = c("BICq", "BICh", "BIC", "BICNgrp"),
  BICq_calc = T,
  logLik_calc = switch(BIC_option[1], BICq = F, T),
  BICq_posterior = NULL,
  checks_complete = F,
  pre_screen = T,
  ranef_keep = NULL,
  lambda.min.full,
  stage1 = F
)
```

## Arguments

| | |
|---|---|
| dat | a list object specifying y (response vector), X (model matrix of all covariates), Z (model matrix for the random effects), and group (numeric factor vector whose value indicates the study, batch, or other group identity to which on observation belongs) |
| offset | This can be used to specify an *a priori* known component to be included in the linear predictor during fitting. Default set to NULL (no offset). If the data argument is NULL, this should be a numeric vector of length equal to the number of cases (the response). If the data argument specifies a data.frame, the offset argument should specify the name of a column in the data.frame. |
| family | a description of the error distribution and link function to be used in the model. Currently, the glmmPen algorithm allows the binomial, gaussian, and poisson families with canonical links only. |
| covar | character string specifying whether the covariance matrix should be unstructured ("unstructured") or diagonal with no covariances between variables ("independent"). Default is set to NULL. If covar is set to NULL and the number of random effects predictors (not including the intercept) is greater than or equal to 10 (i.e. high dimensional), then the algorithm automatically assumes an independent covariance structure and covar is set to "independent". Otherwise if covar is set to NULL and the number of random effects predictors is less than 10, then the algorithm automatically assumes an unstructured covariance structure and covar is set to "unstructured". |
| group_X | vector describing the grouping of the covariates in the model matrix. |
| penalty | character describing the type of penalty to use in the variable selection procedure. Options include 'MCP', 'SCAD', and 'lasso'. Default is MCP penalty. If the random effect covariance matrix is "unstructured", then a group MCP, group SCAD, or group Lasso penalty is used on the random effects coefficients. |
| lambda0_seq | a sequence of non-negative numeric penalty parameters for the fixed and random effect parameters, respectively. If NULL, then a sequence will be automatically calculated. See 'Details' section for more details on these default calculations. |
| lambda1_seq | a sequence of non-negative numeric penalty parameters for the fixed and random effect parameters, respectively. If NULL, then a sequence will be automatically calculated. See 'Details' section for more details on these default calculations. |
| alpha | Tuning parameter for the Mnet estimator which controls the relative contributions from the MCP/SCAD/lasso penalty and the ridge, or L2, penalty. alpha=1 is equivalent to the MCP/SCAD/lasso penalty, while alpha=0 is equivalent to ridge regression. However, alpha=0 is not supported; alpha may be arbibrarily small, but not exactly zero |
| gamma_penalty | The tuning parameter of the MCP and SCAD penalties. Not used by Lasso penalty. Default is 4.0 for SCAD and 3.0 for MCP. |
| trace | an integer specifying print output to include as function runs. Default value is 0. See Details for more information about output provided when trace = 0, 1, or 2. |
| adapt_RW_options | a list of class "adaptControl" from function adaptControl containing the control parameters for the adaptive random walk Metropolis-within-Gibbs procedure. Ignored if optimControl parameter sampler is set to "stan" (default) or "independence". |
| optim_options | a structure of class "optimControl" created from function optimControl that specifies optimization parameters. See the documentation for optimControl for more details on defaults. |

`BIC_option` character string specifing the selection criteria used to select the 'best' model. Default "BICq" option specifies the BIC-ICQ criterion, which requires a fit of a full model; a small penalty of 0.001*(lambda max) is used for the fixed and/or random effects if the number of effects is 10 or greater. The "BICh" option utilizes the hybrid BIC value described in Delattre, Lavielle, and Poursat (2014). The regular "BIC" option penalty term uses (total non-zero coefficients)*(length(y) = total number observations). The "BICNgrp" option penalty term uses (total non-zero coefficients)*(nlevels(group) = number groups).

`BICq_calc` boolean value indicating if the BIC-ICQ criterion should be used to select the best model.

`logLik_calc` logical value specifying if the log likelihood (and log-likelihood based calculations BIC, BICh, and BICNgrp) should be calculated for all of the models in the selection procedure. If BIC-ICQ is used for selection, the log-likelihood is not needed. However, if users are interested in comparing the best models from BIC-ICQ and other BIC-type selection criteria, setting `logLik_calc` to `TRUE` will calculate these other options for all of the models.

`BICq_posterior` an optional character string expressing the path and file basename of a file combination that the will file-back or currently file-backs a `big.matrix` of the posterior draws from the full model. These full model posterior draws will be used in BIC-ICQ calculations if these calculations are requested. If this argument is specified as `NULL` (default) and BIC-ICQ calculations are requested, the posterior draws will be saved in the file combination 'BICq_Posterior_Draws.bin' and 'BICq_Posterior_Draws.desc' in the working directory. See 'Details' section for additional details about the required format of `BICq_posterior` and the file-backed big matrix.

`checks_complete`
boolean value indicating if several data checks have been completed.

`pre_screen` logical value indicating whether pre-screening should be performed before model selection (default `TRUE`). If the number of random effects considered less than 5, no pre-screening will be performed. Pre-screening removes random effects from consideration during the model selection process, which can significantly speed up the algorithm.

`lambda.min.full`
a vector of two numeric values that gives the fixed and random effect penalty values to use in pre-screening and/or the full model fit for the BIC-ICQ calculation (if applicable)

`stage1` boolean value indicating if the first stage of the abbreviated two-stage grid search in the model selection procedure is being performed.

## Value

A list with the following elements:

`results` matrix of summary results for each lambda tuning parameter combination, used to select the 'best' model

`out` list of `fit_dat_B` results for the models with the minimum BICh and minimum BIC values

`coef` matrix of coefficient results for each lambda tuning parameter combination. Rows correspond with the rows of the results matrix.

# Index