# Package ‘coxmeg’

May 5, 2020

**Type** Package

**Title** Cox Mixed-Effects Models for Genome-Wide Association Studies

**Version** 1.0.11

**Date** 2020-05-04

**Author** Liang He

**Maintainer** Liang He <liang.he@duke.edu>

**Description** Fast algorithms for fitting a Cox mixed-effects model for e.g., genome-wide association studies. See Liang He and Alexander Kulminski (2019) <doi:10.1101/729285>.

**License** GPL

**Imports** nloptr, MASS, matrixcalc, SNPRelate, rARPACK, stats, Matrix, methods, utils, coxme, microbenchmark

**LinkingTo** Rcpp, RcppEigen

**Depends** R (>= 3.4), Rcpp (>= 0.12.18)

**RoxygenNote** 6.1.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Repository** CRAN

**Repository/R-Forge/Project** coxmeg

**Repository/R-Forge/Revision** 17

**Repository/R-Forge/DateTimeStamp** 2020-05-05 15:55:01

**Date/Publication** 2020-05-05 20:40:03 UTC

**NeedsCompilation** yes

## R topics documented:

## coxmeg-package *Cox Mixed-Effects Models for Genome-Wide Association Studies*

### Description

Fast algorithms for fitting a Cox mixed-effects model for e.g., genome-wide association studies. See Liang He and Alexander Kulminski (2019) <doi:10.1101/729285>.

### Details

coxmeg is an R package for efficiently conducting GWAS of age-at-onset traits using a Cox mixed-effects model. coxmeg introduces a fast estimation algorithm for general sparse relatedness matrices including but not limited to block-diagonal pedigree-based matrices. coxmeg also introduces a fast and powerful score test for fully dense relatedness matrices, accounting for both population stratification and family structure. In addition, coxmeg can handle positive semidefinite relatedness matrices. Compared to coxme, coxmeg substantially improves the computational efficiency for estimating or testing genetic effects by using a variance component estimated once from a null model, and introducing fast algorithms, including inexact newton methods, preconditioned conjugate gradient methods and stochastic Lanczos quadrature.

### Author(s)

Liang He

Maintainer: Liang He <liang.he@duke.edu>

### References

He. L, Kulminski, A.M., Genome-wide association analysis of age-at-onset traits using Cox mixed-effects models. 2019

### Examples

```
library(Matrix)
library(MASS)
library(coxmeg)

## simulate a block-diagonal relatedness matrix
tau_var <- 0.2
n_f <- 100
mat_list <- list()
size <- rep(10,n_f)
offd <- 0.5
for(i in 1:n_f)
{
   mat_list[[i]] <- matrix(offd,size[i],size[i])
   diag(mat_list[[i]]) <- 1
}
```

```
sigma <- as.matrix(bdiag(mat_list))
n <- nrow(sigma)

## simulate random effexts and outcomes
x <- mvrnorm(1, rep(0,n), tau_var*sigma)
myrates <- exp(x-1)
y <- rexp(n, rate = myrates)
cen <- rexp(n, rate = 0.02 )
ycen <- pmin(y, cen)
outcome <- cbind(ycen,as.numeric(y <= cen))

## fit the Cox mixed-effects model
re = coxmeg(outcome,sigma,type='bd',order=1,detap='diagonal')
re
```

# coxmeg *Fit a Cox mixed-effects model*

## Description

coxmeg returns estimates of the variance component, the HRs and p-values for the predictors.

## Usage

```
coxmeg(outcome, corr, type, X = NULL, FID = NULL, eps = 1e-06,
  min_tau = 1e-04, max_tau = 5, order = 1, detap = NULL,
  opt = "bobyqa", solver = NULL, spd = TRUE, verbose = TRUE,
  mc = 100)
```

## Arguments

outcome A matrix contains time (first column) and status (second column). The status is a binary variable (1 for events / 0 for censored).

corr A relatedness matrix. Can be a matrix or a 'dgCMatrix' class in the Matrix package. Must be symmetric positive definite or symmetric positive semidefinite.

type A string indicating the sparsity structure of the relatedness matrix. Should be 'bd' (block diagonal), 'sparse', or 'dense'. See details.

X An optional matrix of the preidctors with fixed effects. Can be quantitative or binary values. Categorical variables need to be converted to dummy variables. Each row is a sample, and the predictors are columns.

FID An optional string vector of family ID. If provided, the data will be reordered according to the family ID.

eps An optional positive scalar indicating the tolerance in the optimization algorithm. Default is 1e-6.

min_tau An optional positive scalar indicating the lower bound in the optimization algorithm for the variance component tau. Default is 1e-4.

`max_tau` An optional positive scalar indicating the upper bound in the optimization algorithm for the variance component `tau` Default is 5.

`order` An optional integer starting from 0. Only valid when `dense=FALSE`. It specifies the order of approximation used in the inexact newton method. Default is 1.

`detap` An optional string indicating whether to use approximation for log-determinant. Can be 'exact', 'diagonal' or 'slq'. Default is NULL, which lets the function select a method based on 'type' and other information. See details.

`opt` An optional logical scalar for the Optimization algorithm for tau. Can have the following values: 'bobyqa', 'Brent' or 'NM'. Default is 'bobyqa'.

`solver` An optional bianry value that can be either 1 (Cholesky Decomposition using RcppEigen), 2 (PCG) or 3 (Cholesky Decomposition using Matrix). Default is NULL, which lets the function select a solver. See details.

`spd` An optional logical value indicating whether the relatedness matrix is symmetric positive definite. Default is TRUE. See details.

`verbose` An optional logical scalar indicating whether to print additional messages. Default is TRUE.

`mc` An optional integer scalar specifying the number of Monte Carlo samples used for approximating the log-determinant. Only valid when `dense=TRUE` and `detap='slq'`. Default is 100.

## Value

beta: The estimated coefficient for each predictor in X.

HR: The estimated HR for each predictor in X.

sd_beta: The estimated standard error of beta.

p: The p-value.

iter: The number of iterations until convergence.

tau: The estimated variance component.

int_ll: The marginal likelihood (-2*log(lik)) of tau evaluated at the estimate of tau.

rank: The rank of the relatedness matrix.

nsam: Actual sample size.

## About `type`

'bd' is used for a block-diagonal relatedness matrix, or a sparse matrix the inverse of which is also sparse. 'sparse' is used for a general sparse relatedness matrix the inverse of which is not sparse.

## About `spd`

When `spd=TRUE`, the relatedness matrix is treated as SPD. If the matrix is SPSD or not sure, use `spd=FALSE`.

### About `solver`

When `solver=1,3`/`solver=2`, Cholesky decompositon/PCG is used to solve the linear system. When `solver=3`, the solve function in the Matrix package is used, and when `solver=1`, it uses RcppEigen:LDLT to solve linear systems. When `type='dense'`, it is recommended to set `solver=2` to have better computational performance.

### About `detap`

When `detap='exact'`, the exact log-determinant is computed for estimating the variance component. Specifying `detap='diagonal'` uses diagonal approximation, and is only effective for a sparse relatedness matrix. Specifying `detap='slq'` uses stochastic lanczos quadrature approximation.

### Examples

```
library(Matrix)
library(MASS)
library(coxmeg)

## simulate a block-diagonal relatedness matrix
tau_var <- 0.2
n_f <- 100
mat_list <- list()
size <- rep(10,n_f)
offd <- 0.5
for(i in 1:n_f)
{
  mat_list[[i]] <- matrix(offd,size[i],size[i])
  diag(mat_list[[i]]) <- 1
}
sigma <- as.matrix(bdiag(mat_list))
n <- nrow(sigma)

## simulate random effects and outcomes
x <- mvrnorm(1, rep(0,n), tau_var*sigma)
myrates <- exp(x-1)
y <- rexp(n, rate = myrates)
cen <- rexp(n, rate = 0.02 )
ycen <- pmin(y, cen)
outcome <- cbind(ycen,as.numeric(y <= cen))

## fit a Cox mixed-effects model
re = coxmeg(outcome,sigma,type='bd',detap='diagonal',order=1)
re
```

## `coxmeg_m` *Fit a Cox mixed-effects model for estimating HRs for a set of predictors*

## Description

coxmeg_m first estimates the variance component under a null model with only cov, and then analyzing each predictor in X one by one.

## Usage

```
coxmeg_m(X, outcome, corr, type, FID = NULL, cov = NULL, tau = NULL,
  min_tau = 1e-04, max_tau = 5, eps = 1e-06, order = NULL,
  detap = NULL, opt = "bobyqa", score = FALSE, threshold = 0,
  solver = NULL, spd = TRUE, verbose = TRUE, mc = 100)
```

## Arguments

| | |
|---|---|
| X | A matrix of the preidctors. Can be quantitative or binary values. Categorical variables need to be converted to dummy variables. Each row is a sample, and the predictors are columns. |
| outcome | A matrix contains time (first column) and status (second column). The status is a binary variable (1 for failure / 0 for censored). |
| corr | A relatedness matrix. Can be a matrix or a 'dgCMatrix' class in the Matrix package. Must be symmetric positive definite or symmetric positive semidefinite. |
| type | A string indicating the sparsity structure of the relatedness matrix. Should be 'bd' (block diagonal), 'sparse', or 'dense'. See details. |
| FID | An optional string vector of family ID. If provided, the data will be reordered according to the family ID. |
| cov | An optional matrix of the covariates included in the null model for estimating the variance component. Can be quantitative or binary values. Categorical variables need to be converted to dummy variables. Each row is a sample, and the covariates are columns. |
| tau | An optional positive value for the variance component. If tau is given, the function will skip estimating the variance component, and use the given tau to analyze the predictors. |
| min_tau | An optional positive value indicating the lower bound in the optimization algorithm for the variance component tau. Default is 1e-4. |
| max_tau | An optional positive value indicating the upper bound in the optimization algorithm for the variance component tau. Default is 5. |
| eps | An optional positive value indicating the tolerance in the optimization algorithm. Default is 1e-6. |
| order | An optional integer value starting from 0. Only valid when dense=FALSE. It specifies the order of approximation used in the inexact newton method. Default is NULL, which lets coxmeg choose an optimal order. |
| detap | An optional string indicating whether to use approximation for log-determinant. Can be 'exact', 'diagonal' or 'slq'. Default is NULL, which lets the function select a method based on 'type' and other information. See details. |
| opt | An optional logical value for the Optimization algorithm for tau. Can have the following values: 'bobyqa', 'Brent' or 'NM'. Default is 'bobyqa'. |

`score` An optional logical value indicating whether to perform a score test. Default is FALSE.

`threshold` An optional non-negative value. If threshold>0, coxmeg_m will reestimate HRs for those SNPs with a p-value<threshold by first estimating a variant-specific variance component. Default is 0.

`solver` An optional bianry value that can be either 1 (Cholesky Decomposition using RcppEigen), 2 (PCG) or 3 (Cholesky Decomposition using Matrix). Default is NULL, which lets the function select a solver. See details.

`spd` An optional logical value indicating whether the relatedness matrix is symmetric positive definite. Default is TRUE. See details.

`verbose` An optional logical value indicating whether to print additional messages. Default is TRUE.

`mc` An optional integer value specifying the number of Monte Carlo samples used for approximating the log-determinant. Only valid when `dense=TRUE` and `detap='slq'`. Default is 100.

## Value

beta: The estimated coefficient for each predictor in X.

HR: The estimated HR for each predictor in X.

sd_beta: The estimated standard error of beta.

p: The p-value of each SNP.

tau: The estimated variance component.

iter: The number of iterations until convergence.

## About `type`

'bd' is used for a block-diagonal relatedness matrix, or a sparse matrix the inverse of which is also sparse. 'sparse' is used for a general sparse relatedness matrix the inverse of which is not sparse.

## About `spd`

When `spd=TRUE`, the relatedness matrix is treated as SPD. If the matrix is SPSD or not sure, use `spd=FALSE`.

## About `solver`

When `solver=1,3`/`solver=2`, Cholesky decompositon/PCG is used to solve the linear system. When `solver=3`, the solve function in the Matrix package is used, and when `solver=1`, it uses RcppEigen:LDLT to solve linear systems. When `type='dense'`, it is recommended to set `solver=2` to have better computational performance.

## About `detap`

When `detap='exact'`, the exact log-determinant is computed for estimating the variance component. Specifying `detap='diagonal'` uses diagonal approximation, and is only effective for a sparse relatedness matrix. Specifying `detap='slq'` uses stochastic lanczos quadrature approximation.

## Examples

```
library(Matrix)
library(MASS)
library(coxmeg)

## simulate a block-diagonal relatedness matrix
tau_var <- 0.2
n_f <- 100
mat_list <- list()
size <- rep(10,n_f)
offd <- 0.5
for(i in 1:n_f)
{
  mat_list[[i]] <- matrix(offd,size[i],size[i])
  diag(mat_list[[i]]) <- 1
}
sigma <- as.matrix(bdiag(mat_list))
n <- nrow(sigma)

## simulate random effects and outcomes
x <- mvrnorm(1, rep(0,n), tau_var*sigma)
myrates <- exp(x-1)
y <- rexp(n, rate = myrates)
cen <- rexp(n, rate = 0.02 )
ycen <- pmin(y, cen)
outcome <- cbind(ycen,as.numeric(y <= cen))

## simulate genotypes
g = matrix(rbinom(n*5,2,0.5),n,5)

## The following command will first estimate the variance component without g,
## and then use it to estimate the HR for each preditor in g.
re = coxmeg_m(g,outcome,sigma,type='bd',tau=0.5,detap='diagonal',order=1)
re
```

---

# coxmeg_plink *Perform GWAS using a Cox mixed-effects model with plink files as input*

---

## Description

coxmeg_plink first estimates the variance component under a null model with only cov if tau is not given, and then analyzing each SNP in the plink files.

## Usage

```
coxmeg_plink(pheno, corr, type, bed = NULL, tmp_dir = NULL,
  cov_file = NULL, tau = NULL, maf = 0.05, min_tau = 1e-04,
  max_tau = 5, eps = 1e-06, order = NULL, detap = NULL,
  opt = "bobyqa", score = FALSE, threshold = 0, solver = NULL,
  spd = TRUE, mc = 100, verbose = TRUE)
```

## Arguments

pheno
: A string value indicating the file name or the full path of a pheno file. The files must be in the working directory if the full path is not given. The file is in plink pheno format, containing the following four columns, family ID, individual ID, time and status. The status is a binary variable (1 for events/0 for censored).

corr
: A relatedness matrix. Can be a matrix or a 'dgCMatrix' class in the Matrix package. Must be symmetric positive definite or symmetric positive semidefinite.

type
: A string indicating the sparsity structure of the relatedness matrix. Should be 'bd' (block diagonal), 'sparse', or 'dense'. See details.

bed
: A optional string value indicating the file name or the full path of a plink bed file (without .bed). The files must be in the working directory if the full path is not given. If not provided, only the variance component will be returned.

tmp_dir
: A optional directory to store temporary .gds files. The directory needs to be specified when bed is provided.

cov_file
: An optional string value indicating the file name or the full path of a covariate file. The files must be in the working directory if the full path is not given. Same as the cov file in plink, the first two columns are family ID and individual ID. The covariates are included in the null model for estimating the variance component. The covariates can be quantitative or binary values. Categorical variables need to be converted to dummy variables.

tau
: An optional positive value for the variance component. If tau is given, the function will skip estimating the variance component, and use the given tau to analyze the SNPs.

maf
: An optional positive value. All SNPs with MAF<maf in the bed file will not be analyzed. Default is 0.05.

min_tau
: An optional positive value indicating the lower bound in the optimization algorithm for the variance component tau. Default is 1e-4.

max_tau
: An optional positive value indicating the upper bound in the optimization algorithm for the variance component tau. Default is 5.

eps
: An optional positive value indicating the tolerance in the optimization algorithm. Default is 1e-6.

order
: An optional integer value starting from 0. Only effective when `dense=FALSE`. It specifies the order of approximation used in the inexact newton method. Default is NULL, which lets coxmeg choose an optimal order.

detap
: An optional string indicating whether to use approximation for log-determinant. Can be 'exact', 'diagonal' or 'slq'. Default is NULL, which lets the function select a method based on 'type' and other information. See details.

opt
: An optional string value for the Optimization algorithm for tau. Can have the following values: 'bobyqa', 'Brent' or 'NM'. Default is 'bobyqa'.

score
: An optional logical value indicating whether to perform a score test. Default is FALSE.

| | |
|---|---|
| `threshold` | An optional non-negative value. If threshold>0, coxmeg_m will reestimate HRs for those SNPs with a p-value<threshold by first estimating a variant-specific variance component. Default is 0. |
| `solver` | An optional binary value taking either 1 or 2. Default is 1. See details. |
| `spd` | An optional logical value indicating whether the relatedness matrix is symmetric positive definite. Default is TRUE. See details. |
| `mc` | An optional integer value specifying the number of Monte Carlo samples used for approximating the log-determinant. Only valid when `dense=TRUE` and `detap='slq'`. Default is 100. |
| `verbose` | An optional logical value indicating whether to print additional messages. Default is TRUE. |

## Value

beta: The estimated coefficient for each predictor in X.

HR: The estimated HR for each predictor in X.

sd_beta: The estimated standard error of beta.

p: The p-value of each SNP.

tau: The estimated variance component.

rank: The rank of the relatedness matrix.

nsam: Actual sample size.

## About `type`

'bd' is used for a block-diagonal relatedness matrix, or a sparse matrix the inverse of which is also sparse. 'sparse' is used for a general sparse relatedness matrix the inverse of which is not sparse.

## About `corr`

The subjects in `corr` must be in the same order as in the plink fam file.

## About missing values

`pheno` -9 for missing values, `cov_file` NA for missing values.

## About temporary files

The function will create a temporary gds file with approximately the same size as the bed file. The temporary file will be removed when the analysis is done.

## About `spd`

When `spd=TRUE`, the relatedness matrix is treated as SPD. If the matrix is SPSD or not sure, set `spd=FALSE`.

### About solver

When solver=1,3/solver=2, Cholesky decompositon/PCG is used to solve the linear system. When solver=3, the solve function in the Matrix package is used, and when solver=1, it uses RcppEigen:LDLT to solve linear systems. When type='dense', it is recommended to set solver=2 to have better computational performance.

### About detap

When detap='exact', the exact log-determinant is computed for estimating the variance component. Specifying detap='diagonal' uses diagonal approximation, and is only effective for a sparse relatedness matrix. Specifying detap='slq' uses stochastic lanczos quadrature approximation.

### Examples

```
library(Matrix)
library(MASS)
library(coxmeg)

## build a block-diagonal relatedness matrix
n_f <- 600
mat_list <- list()
size <- rep(5,n_f)
offd <- 0.5
for(i in 1:n_f)
{
  mat_list[[i]] <- matrix(offd,size[i],size[i])
  diag(mat_list[[i]]) <- 1
}
sigma <- as.matrix(bdiag(mat_list))

## Estimate variance component under a null model
pheno = system.file("extdata", "ex_pheno.txt", package = "coxmeg")
cov = system.file("extdata", "ex_cov.txt", package = "coxmeg")
bed = system.file("extdata", "example_null.bed", package = "coxmeg")
bed = substr(bed,1,nchar(bed)-4)
re = coxmeg_plink(pheno,sigma,type='bd',bed=bed,tmp_dir=tempdir(),cov_file=cov,
detap='diagonal',order=1)
re
```

---

## fit_ppl — *Estimate HRs using PPL given a known variance component (tau)*

---

### Description

fit_ppl returns estimates of HRs and their p-values given a known variance component (tau).

### Usage

```
fit_ppl(X, outcome, corr, type, tau = 0.5, FID = NULL, eps = 1e-06,
  order = 1, solver = NULL, spd = TRUE, verbose = TRUE)
```

### Arguments

`X` A matrix of the preidctors. Can be quantitative or binary values. Categorical variables need to be converted to dummy variables. Each row is a sample, and the predictors are columns.

`outcome` A matrix contains time (first column) and status (second column). The status is a binary variable (1 for failure / 0 for censored).

`corr` A relatedness matrix. Can be a matrix or a 'dgCMatrix' class in the Matrix package. Must be symmetric positive definite or symmetric positive semidefinite.

`type` A string indicating the sparsity structure of the relatedness matrix. Should be 'bd' (block diagonal), 'sparse', or 'dense'. See details.

`tau` A positive scalar. A variance component given by the user. Default is 0.5.

`FID` An optional string vector of family ID. If provided, the data will be reordered according to the family ID.

`eps` An optional positive value indicating the tolerance in the optimization algorithm. Default is 1e-6.

`order` An optional integer value starting from 0. Only valid when dense=FALSE. It specifies the order of approximation used in the inexact newton method. Default is 1.

`solver` An optional bianry value that can be either 1 (Cholesky Decomposition using RcppEigen), 2 (PCG) or 3 (Cholesky Decomposition using Matrix). Default is NULL, which lets the function select a solver. See details.

`spd` An optional logical value indicating whether the relatedness matrix is symmetric positive definite. Default is TRUE.

`verbose` An optional logical value indicating whether to print additional messages. Default is TRUE.

### Value

beta: The estimated coefficient for each predictor in X.

HR: The estimated HR for each predictor in X.

sd_beta: The estimated standard error of beta.

p: The p-value.

iter: The number of iterations until convergence.

ppl: The PPL when the convergence is reached.

### About `type`

'bd' is used for a block-diagonal relatedness matrix, or a sparse matrix the inverse of which is also sparse. 'sparse' is used for a general sparse relatedness matrix the inverse of which is not sparse.

### About `solver`

When `solver=1,3`/`solver=2`, Cholesky decompositon/PCG is used to solve the linear system. When `solver=3`, the solve function in the Matrix package is used, and when `solver=1`, it uses RcppEigen:LDLT to solve linear systems.

### Examples

```
library(Matrix)
library(MASS)
library(coxmeg)

## simulate a block-diagonal relatedness matrix
tau_var <- 0.2
n_f <- 100
mat_list <- list()
size <- rep(10,n_f)
offd <- 0.5
for(i in 1:n_f)
{
  mat_list[[i]] <- matrix(offd,size[i],size[i])
  diag(mat_list[[i]]) <- 1
}
sigma <- as.matrix(bdiag(mat_list))
n <- nrow(sigma)

## simulate random effexts and outcomes
x <- mvrnorm(1, rep(0,n), tau_var*sigma)
myrates <- exp(x-1)
y <- rexp(n, rate = myrates)
cen <- rexp(n, rate = 0.02 )
ycen <- pmin(y, cen)
outcome <- cbind(ycen,as.numeric(y <= cen))

## fit the ppl
re = fit_ppl(x,outcome,sigma,type='bd',tau=0.5,order=1)
re
```

# Index