Package 'GMMAT'

January 20, 2025

Version 1.4.2

Date 2023-11-17

Title Generalized Linear Mixed Model Association Tests

Description Perform association tests using generalized linear mixed models (GLMMs) in genome-wide association studies (GWAS) and sequencing association studies. First, GMMAT fits a GLMM with covariate adjustment and random effects to account for population structure and familial or cryptic relatedness. For GWAS, GMMAT performs score tests for each genetic variant as proposed in Chen et al. (2016) <DOI:10.1016/j.ajhg.2016.02.012>. For candidate gene studies, GMMAT can also perform Wald tests to get the effect size estimate for each genetic variant. For rare variant analysis from sequencing association studies, GMMAT performs the variant Set Mixed Model Association Tests (SMMAT) as proposed in Chen et al. (2019) <DOI:10.1016/j.ajhg.2018.12.012>, including the burden test, the sequence kernel association test (SKAT), SKAT-O and an efficient hybrid test of the burden test and SKAT, based on user-defined variant sets.

License GPL (>= 3)

Copyright See COPYRIGHTS for details.

Imports Rcpp, CompQuadForm, foreach, parallel, Matrix, methods, data.table

Suggests doMC, SeqArray, SeqVarTools, testthat

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

NeedsCompilation yes

Depends R (>= 3.2.0)

Author Han Chen [aut, cre],
Matthew Conomos [aut],
Duy Pham [aut],
Arthor Gilly [ctb],
Robert Gentleman [ctb, cph] (Author and copyright holder of the C function Brent_fmin),
Ross Ihaka [ctb, cph] (Author and copyright holder of the C function Brent_fmin),
The R Core Team [ctb, cph] (Author and copyright holder of the C

function Brent_fmin),
The R Foundation [cph] (Copyright holder of the C function Brent_fmin),
Eric Biggers [ctb, cph] (Author and copyright holder of included
libdeflate library),
Tino Reichardt [ctb, cph] (Author and copyright holder of threading
code used in the included Zstandard (zstd) library),
Meta Platforms, Inc. and affiliates [cph] (Copyright holder of included
Zstandard (zstd) library)

Maintainer Han Chen <han.chen.2@uth.tmc.edu>

Repository CRAN

Date/Publication 2023-11-17 10:20:33 UTC

Contents

`GMMAT-package` *Generalized Linear Mixed Model Association Tests*

Description

An R package for performing association tests using generalized linear mixed models (GLMMs) in genome-wide association studies (GWAS) and sequencing association studies. First, GMMAT fits a GLMM with covariate adjustment and random effects to account for population structure and familial or cryptic relatedness. For GWAS, GMMAT performs score tests for each genetic variant. For candidate gene studies, GMMAT can also perform Wald tests to get the effect size estimate for each genetic variant. For rare variant analysis from sequencing association studies, GMMAT performs the variant Set Mixed Model Association Tests (SMMAT), including the burden test, the sequence kernel association test (SKAT), SKAT-O and an efficient hybrid test of the burden test and SKAT, based on user-defined variant sets.

Details

Package:	GMMAT
Type:	Package
Version:	1.4.2
Date:	2023-11-17

License: GPL (>= 3)

Author(s)

Han Chen, Matthew P. Conomos, Duy T. Pham

Maintainer: Han Chen <Han.Chen.2@uth.tmc.edu>

References

Brent, R.P. (1973) "Chapter 4: An Algorithm with Guaranteed Convergence for Finding a Zero of a Function", Algorithms for Minimization without Derivatives, Englewood Cliffs, NJ: Prentice-Hall, ISBN 0-13-022335-2.

Breslow, N.E. and Clayton, D.G. (1993) Approximate Inference in Generalized Linear Mixed Models. Journal of the American Statistical Association 88, 9-25.

Chen, H., Huffman, J.E., Brody, J.A., Wang, C., Lee, S., Li, Z., Gogarten, S.M., Sofer, T., Bielak, L.F., Bis, J.C., et al. (2019) Efficient variant set mixed model association tests for continuous and binary traits in large-scale whole-genome sequencing studies. The American Journal of Human Genetics 104, 260-274.

Chen, H., Wang, C., Conomos, M.P., Stilp, A.M., Li, Z., Sofer, T., Szpiro, A.A., Chen, W., Brehm, J.M., Celedón, J.C., Redline, S., Papanicolaou, G.J., Thornton, T.A., Laurie, C.C., Rice, K. and Lin, X. (2016) Control forpopulation structure and relatedness for binary traits in genetic association studies via logistic mixed models. The American Journal of Human Genetics 98, 653-666.

Gilmour, A.R., Thompson, R. and Cullis, B.R. (1995) Average Information REML: An Efficient Algorithm for Variance Parameter Estimation in Linear Mixed Models. Biometrics 51, 1440-1450.

Lee, S., Teslovich, T., Boehnke, M., Lin, X. (2013) General framework for meta-analysis of rare variants in sequencing association studies. The American Journal of Human Genetics 93, 42-53.

Lee, S., Wu, M.C., Lin, X. (2012) Optimal tests for rare variant effects in sequencing association studies. Biostatistics 13, 762-775.

Nelder, J.A. and Mead, R. (1965) A simplex algorithm for function minimization. Computer Journal 7, 308-313.

Sun, J., Zheng, Y., Hsu, L. (2013) A unified mixed-effects model for rare-variant association in sequencing studies. Genetic Epidemiology 37, 334-344.

Wu, M.C., Lee, S., Cai, T., Li, Y., Boehnke, M., Lin, X. (2011) Rare-variant association testing for sequencing data with the sequence kernel association test. The American Journal of Human Genetics 89, 82-93.

Yang, J., Lee, S.H., Goddard, M.E. and Visscher, P.M. (2011) GCTA: A Tool for Genome-wide Complex Trait Analysis. The American Journal of Human Genetics 88, 76-82.

Zhou, X. and Stephens, M. (2012) Genome-wide efficient mixed-model analysis for association studies. Nature Genetics 44, 821-824.

example *Example dataset*

Description

Example dataset for GMMAT.

Format

Contains the following objects:

pheno a data frame of 400 observations from a cross-sectional study with 5 variables: id, disease, trait, age and sex.

pheno2 a data frame of 2,000 observations from a longitudinal study with 400 individuals and 5 variables: id, y.repeated, y.trend, time and sex.

GRM a genetic relationship matrix for 400 observations.

glmm.score *Performing GLMM based score tests*

Description

Use a glmmkin class object from the null GLMM to perform score tests for association with genotypes in a plink .bed file (binary genotypes), a GDS file .gds, or a plain text file (or compressed .gz or .bz2 file).

Usage

```
glmm.score(obj, infile, outfile, BGEN.samplefile = NULL, center = T, select = NULL,
MAF.range = c(1e-7, 0.5), miss.cutoff = 1,
missing.method = "impute2mean", nperbatch = 100, tol = 1e-5,
infile.nrow = NULL, infile.nrow.skip = 0, infile.sep = "\t",
infile.na = "NA", infile.ncol.skip = 1, infile.ncol.print = 1,
infile.header.print = "SNP", is.dosage = FALSE, ncores = 1, verbose = FALSE)
```

Arguments

obj	a class glmmkin or class glmmkin.multi object, returned by fitting the null GLMM using glmmkin.
infile	the input file name or an object of class SeqVarGDSClass. Note that for plink binary genotype files only the prefix without .bed, .bim or .fam should be used. Only SNP major mode recognized in the binary file. Alternatively, it can be the full name of a BGEN file (including the suffix .bgen), a GDS file (including the suffix .gds), or a plain text file with some delimiters (comma, space, tab or something else), with one row for each SNP and one column for each individual.

In that case, SNPs should be coded as numeric values (0/1/2 or dosages allowed, A/C/G/T coding is not recognized). There can be additional rows and columns to skip at the beginning. The order of individuals can be different from `obj` in the null GLMM (see the argument `select`). Some compressed files (.gz and .bz2) also allowed. If `infile` is an object of class SeqVarGDSClass, the .gds file will be closed upon successful completion of the function.

`outfile` the output file name.

`BGEN.samplefile`
path to the BGEN sample file. Required when the BGEN file does not contain sample identifiers or the `select` parameter is NULL (default = NULL).

`center` a logical switch for centering genotypes before tests. If TRUE, genotypes will be centered to have mean 0 before tests, otherwise raw values will be directly used in tests (default = TRUE).

`select` an optional vector indicating the order of individuals in `infile`. If supplied, the length must match the number of individuals in `infile` (default = NULL). Individuals to be excluded should be coded 0. For example, `select = c(2, 3, 1, 0)` means the 1st individual in `infile` corresponds to the 2nd individual in `obj`, the 2nd individual in `infile` corresponds to the 3rd individual in `obj`, the 3rd individual in `infile` corresponds to the 1st individual in `obj`, the 4th individual in `infile` is not included in `obj`. If there are any duplicated `id_include` in `obj` (longitudinal data analysis), indices in `select` should match the order of individuals with unique `id_include` in `obj`. For plink binary genotype files and GDS files, this argument is not required and the sample ID's are automatically matched.

`MAF.range` a numeric vector of length 2 defining the minimum and maximum minor allele frequencies of variants that should be included in the analysis (default = c(1e-7, 0.5)).

`miss.cutoff` the maximum missing rate allowed for a variant to be included (default = 1, including all variants).

`missing.method` method of handling missing genotypes. Either "impute2mean" or "omit" (default = "impute2mean").

`nperbatch` an integer for how many SNPs should be tested in a batch (default = 100). The computational time can increase dramatically if this value is either small or large. The optimal value for best performance depends on the user's system.

`tol` the threshold for determining monomorphism. If a SNP has value range less than the tolerance, it will be considered monomorphic and its association test p-value will be NA (default = 1e-5). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.nrow` number of rows to read in `infile`, including number of rows to skip at the beginning. If NULL, the program will determine how many rows there are in `infile` automatically and read all rows (default = NULL). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.nrow.skip`
number of rows to skip at the beginning of `infile`. Must be nonnegative integers. Useful when header or comment lines are present (default = 0). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.sep` delimiter in `infile` (default = "\t"). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.na` symbol in `infile` to denote missing genotypes (default = "NA"). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.ncol.skip` number of columns to skip before genotype data in `infile`. These columns can be SNP name, alleles and/or quality measures and should be placed at the beginning in each line. After skipping these columns, the program will read in genotype data and perform score tests. Must be nonnegative integers. It is recommended that SNP name should be included as the first column in `infile` and genotype data should start from the second column or later (default = 1). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.ncol.print` a vector indicating which column(s) in `infile` should be printed to the output directly. These columns can be SNP name, alleles and/or quality measures placed at the beginning in each line. Must be nonnegative integers, no greater than `infile.ncol.skip` and sorted numerically in ascending order. By default, it is assumed that the first column is SNP name and genotype data start from the second column, and SNP name should be carried over to the output (default = 1). Should be set to NULL if `infile.ncol.skip` is 0. Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.header.print` a character vector indicating column name(s) of column(s) selected to print by `infile.ncol.print` (default = "SNP"). Should be set to NULL if `infile.ncol.skip` is 0. Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`is.dosage` a logical switch for whether imputed dosage should be used from a GDS `infile` (default = FALSE).

`ncores` a positive integer indicating the number of cores to be used in parallel computing (default = 1).

`verbose` a logical switch for whether a progress bar should be shown for a GDS `infile` (default = FALSE).

Value

NULL if `infile` is a BGEN file (.bgen) or a GDS file (.gds), otherwise computational time in seconds, excluding I/O time.

Author(s)

Han Chen, Duy T. Pham

References

Chen, H., Wang, C., Conomos, M.P., Stilp, A.M., Li, Z., Sofer, T., Szpiro, A.A., Chen, W., Brehm, J.M., Celedón, J.C., Redline, S., Papanicolaou, G.J., Thornton, T.A., Laurie, C.C., Rice, K. and Lin, X. (2016) Control forpopulation structure and relatedness for binary traits in genetic association studies via logistic mixed models. The American Journal of Human Genetics 98, 653-666.

See Also

glmmkin, glmm.wald

Examples

```
data(example)
attach(example)
model0 <- glmmkin(disease ~ age + sex, data = pheno, kins = GRM, id = "id",
       family = binomial(link = "logit"))
plinkfiles <- strsplit(system.file("extdata", "geno.bed", package = "GMMAT"),
       ".bed", fixed = TRUE)[[1]]
outfile.bed <- tempfile()
glmm.score(model0, infile = plinkfiles, outfile = outfile.bed)
if(requireNamespace("SeqArray", quietly = TRUE) && requireNamespace("SeqVarTools",
        quietly = TRUE)) {
infile <- system.file("extdata", "geno.gds", package = "GMMAT")
outfile.gds <- tempfile()
glmm.score(model0, infile = infile, outfile = outfile.gds)
unlink(outfile.gds)
}
infile <- system.file("extdata", "geno.txt", package = "GMMAT")
outfile.text <- tempfile()
glmm.score(model0, infile = infile, outfile = outfile.text, infile.nrow.skip = 5,
infile.ncol.skip = 3, infile.ncol.print = 1:3,
infile.header.print = c("SNP", "Allele1", "Allele2"))
infile <- system.file("extdata", "geno.bgen", package = "GMMAT")
samplefile <- system.file("extdata", "geno.sample", package = "GMMAT")
outfile.bgen <- tempfile()
glmm.score(model0, infile = infile, BGEN.samplefile = samplefile,
        outfile = outfile.bgen)
unlink(c(outfile.bed, outfile.text, outfile.bgen))
```

---

glmm.score.meta *Performing meta-analysis for GLMM based score test results*

Description

Use output files from GLMM based score tests to perform meta-analysis.

Usage

```
glmm.score.meta(files, outfile, SNP = rep("SNP", length(files)),
A1 = rep("A1", length(files)), A2 = rep("A2", length(files)))
```

Arguments

files	a vector of input file names. The input files should be the output files of glmm.score( ), or customized tab or space delimited files that include at least 8 columns: SNP, effect allele, noneffect allele, N, AF, SCORE, VAR and PVAL. The column names of SNP, effect allele and noneffect allele can be customized and provided in SNP, A1 and A2.
outfile	the output file name.
SNP	a character vector of SNP column names in each input file. The length and order must match the length and order of files (default = rep("SNP", length(files))).
A1	a character vector of allele 1 column names in each input file. The length and order must match the length and order of files (default = rep("A1", length(files))). Note that glmm.score.meta( ) does not define A1 as the effect allele or noneffect allele: it is the user's choice. However, the choice should be consistent across different studies, if A1 column is the effect allele in one study but the noneffect allele in another, meta-analysis results will be incorrect.
A2	a character vector of allele 2 column names in each input file. The length and order must match the length and order of files (default = rep("A2", length(files))). Note that glmm.score.meta( ) does not define A2 as the effect allele or noneffect allele: it is the user's choice. However, the choice should be consistent across different studies, if A2 column is the effect allele in one study but the noneffect allele in another, meta-analysis results will be incorrect.

Value

a data frame containing the following:

SNP	SNP name.
A1	allele 1.
A2	allele 2.
N	total sample size.
AF	effect allele frequency (user-defined: can be either allele 1 or allele 2).
SCORE	the summary score of the effect allele.
VAR	the variance of the summary score.
PVAL	meta-analysis p-value.

Author(s)

Han Chen

See Also

glmm.score

Examples

```
infile1 <- system.file("extdata", "meta1.txt", package = "GMMAT")
infile2 <- system.file("extdata", "meta2.txt", package = "GMMAT")
infile3 <- system.file("extdata", "meta3.txt", package = "GMMAT")
outfile <- tempfile()
glmm.score.meta(files = c(infile1, infile2, infile3), outfile = outfile,
SNP = rep("SNP", 3), A1 = rep("A1", 3), A2 = rep("A2", 3))
unlink(outfile)
```

`glmm.wald` *Performing GLMM based Wald tests*

Description

Fit a GLMM under the alternative hypothesis to perform Wald tests for association with genotypes in a plink .bed file (binary genotypes), a GDS file .gds, or a plain text file (or compressed .gz or .bz2 file).

Usage

```
glmm.wald(fixed, data = parent.frame(), kins = NULL, id, random.slope = NULL,
groups = NULL, family = binomial(link = "logit"), infile, snps,
method = "REML", method.optim = "AI", maxiter = 500, tol = 1e-5,
taumin = 1e-5, taumax = 1e5, tauregion = 10, center = T,
select = NULL, missing.method = "impute2mean", infile.nrow = NULL,
infile.nrow.skip = 0, infile.sep = "\t", infile.na = "NA",
snp.col = 1, infile.ncol.skip = 1, infile.ncol.print = 1,
infile.header.print = "SNP", is.dosage = FALSE, verbose = FALSE, ...)
```

Arguments

`fixed`	an object of class `formula` (or one that can be coerced to that class): a symbolic description of the fixed effects model (without including any `snps` to be tested) to be fitted.
`data`	a data frame or list (or object coercible by `as.data.frame` to a data frame) containing the variables in the model.
`kins`	a known positive semi-definite relationship matrix (e.g. kinship matrix in genetic association studies) or a list of known positive semi-definite relationship matrices. The rownames and colnames of these matrices must at least include all samples as specified in the `id` column of the data frame `data`. If not provided, glmmkin will switch to the generalized linear model with no random effects (default = NULL).
`id`	a column in the data frame `data`, indicating the id of samples. When there are duplicates in id, the data is assumed to be longitudinal with repeated measures.

`random.slope` an optional column indicating the random slope for time effect used in a mixed effects model for cross-sectional data with related individuals, and longitudinal data. It must be included in the names of `data`. There must be duplicates in `id` and `method.optim` must be "AI" (default = NULL).

`groups` an optional categorical variable indicating the groups used in a heteroscedastic linear mixed model (allowing residual variances in different groups to be different). This variable must be included in the names of `data`, and `family` must be "gaussian" and `method.optim` must be "AI" (default = NULL).

`family` a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See `family` for details of family functions.)

`infile` the input file name. Note that for plink binary genotype files only the prefix without .bed, .bim or .fam should be used. Only SNP major mode recognized in the binary file. Alternatively, it can be the full name of a GDS file (including the suffix .gds) or a plain text file with some delimiters (comma, space, tab or something else), with one row for each SNP and one column for each individual. In that case, SNPs should be coded as numeric values (0/1/2 or dosages allowed, A/C/G/T coding is not recognized). There can be additional rows and columns to skip at the beginning. The order of individuals can be different from `obj` in the null GLMM (see the argument `select`). Some compressed files (.gz and .bz2) also allowed.

`snps` a vector of SNP names to be tested.

`method` method of fitting the generalized linear mixed model. Either "REML" or "ML" (default = "REML").

`method.optim` optimization method of fitting the generalized linear mixed model. Either "AI", "Brent" or "Nelder-Mead" (default = "AI").

`maxiter` a positive integer specifying the maximum number of iterations when fitting the generalized linear mixed model (default = 500).

`tol` a positive number specifying tolerance, the difference threshold for parameter estimates below which iterations should be stopped. Also the threshold for determining monomorphism. If a SNP has value range less than the tolerance, it will be considered monomorphic and its association test p-value will be NA (default = 1e-5).

`taumin` the lower bound of search space for the variance component parameter τ (default = 1e-5), used when `method.optim` = `"Brent"`. See `glmmkin`.

`taumax` the upper bound of search space for the variance component parameter τ (default = 1e5), used when `method.optim` = `"Brent"`. See `glmmkin`.

`tauregion` the number of search intervals for the REML or ML estimate of the variance component parameter τ (default = 10), used when `method.optim` = `"Brent"`. See `glmmkin`.

`center` a logical switch for centering genotypes before tests. If TRUE, genotypes will be centered to have mean 0 before tests, otherwise raw values will be directly used in tests (default = TRUE).

`select` an optional vector indicating the order of individuals in `infile`. If supplied, the length must match the number of individuals in `infile` (default = NULL). Individuals to be excluded should be coded 0. For example, `select = c(2, 3, 1, 0)` means the 1st individual in `infile` corresponds to the 2nd individual in `data`, the 2nd individual in `infile` corresponds to the 3rd individual in `data`, the 3rd individual in `infile` corresponds to the 1st individual in `data`, the 4th individual in `infile` is not included in `data`. If there are any duplicated `id` in `data` (longitudinal data analysis), indices in `select` should match the order of individuals with unique `id` in `data`. For plink binary genotype files and GDS files, this argument is not required and the sample ID's are automatically matched.

`missing.method` method of handling missing genotypes. Either "impute2mean" or "omit" (default = "impute2mean").

`infile.nrow` number of rows to read in `infile`, including number of rows to skip at the beginning. If NULL, the program will determine how many rows there are in `infile` automatically and read all rows (default = NULL). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.nrow.skip`
number of rows to skip at the beginning of `infile`. Must be nonnegative integers. Useful when header or comment lines are present (default = 0). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.sep` delimiter in `infile` (default = "\t"). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.na` symbol in `infile` to denote missing genotypes (default = "NA"). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`snp.col` a positive integer specifying which column in `infile` is SNP names. Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.ncol.skip`
number of columns to skip before genotype data in `infile`. These columns can be SNP name, alleles and/or quality measures and should be placed at the beginning in each line. After skipping these columns, the program will read in genotype data and perform Wald tests. Must be positive integers. It is recommended that SNP name should be included as the first column in `infile` and genotype data should start from the second column or later (default = 1). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.ncol.print`
a vector indicating which column(s) in `infile` should be shown in the results. These columns can be SNP name, alleles and/or quality measures placed at the beginning in each line. Must be positive integers, no greater than `infile.ncol.skip` and sorted numerically in ascending order. By default, it is assumed that the first column is SNP name and genotype data start from the second column, and SNP name should be carried over to the results (default = 1). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

`infile.header.print`
a character vector indicating column name(s) of column(s) selected to print by `infile.ncol.print` (default = "SNP"). Only used when `infile` is a plain text file (or compressed .gz or .bz2 file).

is.dosage	a logical switch for whether imputed dosage should be used from a GDS infile (default = FALSE).
verbose	a logical switch for printing a progress bar and detailed information (parameter estimates in each iteration) for testing and debugging purpose (default = FALSE).
...	additional arguments that could be passed to glm.

Value

if infile is a plain text file, a data frame containing variables included in infile.header.print and the following:

N	number of individuals with non-missing genotypes for each SNP.
AF	effect allele frequency for each SNP.
BETA	effect size estimate for each SNP from the GLMM under the alternative hypothesis.
SE	standard error of the effect size estimate for each SNP.
PVAL	Wald test p-value for each SNP.
converged	a logical indicator for convergence for each SNP.

if infile is the prefix of plink binary files (.bed, .bim and .fam), a data frame containing the following:

CHR	Chromosome, copied from .bim file.
SNP	SNP name, as supplied in snps.
cM	genetic location in centi Morgans, copied from .bim file.
POS	physical position in base pairs, copied from .bim file.
A1	allele 1, copied from .bim file.
A2	allele 2, copied from .bim file.
N	number of individuals with non-missing genotypes for each SNP.
AF	effect allele frequency for each SNP.
BETA	effect size estimate for each SNP from the GLMM under the alternative hypothesis.
SE	standard error of the effect size estimate for each SNP.
PVAL	Wald test p-value for each SNP.
converged	a logical indicator for convergence for each SNP.

if infile is a GDS file (.gds), a data frame containing the following:

SNP	SNP name, as supplied in snps.
CHR	Chromosome, copied from .gds file.
POS	physical position in base pairs, copied from .gds file.
REF	reference allele, copied from .gds file.
ALT	alternate allele, copied from .gds file.

N	number of individuals with non-missing genotypes for each SNP.
AF	ALT allele frequency for each SNP.
BETA	effect size estimate for each SNP from the GLMM under the alternative hypothesis.
SE	standard error of the effect size estimate for each SNP.
PVAL	Wald test p-value for each SNP.
converged	a logical indicator for convergence for each SNP.

Author(s)

Han Chen, Matthew P. Conomos

References

Brent, R.P. (1973) "Chapter 4: An Algorithm with Guaranteed Convergence for Finding a Zero of a Function", Algorithms for Minimization without Derivatives, Englewood Cliffs, NJ: Prentice-Hall, ISBN 0-13-022335-2.

Breslow, N.E. and Clayton, D.G. (1993) Approximate Inference in Generalized Linear Mixed Models. Journal of the American Statistical Association 88, 9-25.

Chen, H., Wang, C., Conomos, M.P., Stilp, A.M., Li, Z., Sofer, T., Szpiro, A.A., Chen, W., Brehm, J.M., Celedón, J.C., Redline, S., Papanicolaou, G.J., Thornton, T.A., Laurie, C.C., Rice, K. and Lin, X. (2016) Control forpopulation structure and relatedness for binary traits in genetic association studies via logistic mixed models. The American Journal of Human Genetics 98, 653-666.

Gilmour, A.R., Thompson, R. and Cullis, B.R. (1995) Average Information REML: An Efficient Algorithm for Variance Parameter Estimation in Linear Mixed Models. Biometrics 51, 1440-1450.

Nelder, J.A. and Mead, R. (1965) A simplex algorithm for function minimization. Computer Journal 7, 308-313.

Yang, J., Lee, S.H., Goddard, M.E. and Visscher, P.M. (2011) GCTA: A Tool for Genome-wide Complex Trait Analysis. The American Journal of Human Genetics 88, 76-82.

Zhou, X. and Stephens, M. (2012) Genome-wide efficient mixed-model analysis for association studies. Nature Genetics 44, 821-824.

See Also

glmmkin, glmm.score

Examples

```
data(example)
attach(example)
snps <- c("SNP10", "SNP25", "SNP1", "SNP0")
plinkfiles <- strsplit(system.file("extdata", "geno.bed", package = "GMMAT"),
       ".bed", fixed = TRUE)[[1]]
glmm.wald(disease ~ age + sex, data = pheno, kins = GRM, id = "id",
family = binomial(link = "logit"), infile = plinkfiles, snps = snps)
if(requireNamespace("SeqArray", quietly = TRUE) && requireNamespace("SeqVarTools",
        quietly = TRUE)) {
```

```
infile <- system.file("extdata", "geno.gds", package = "GMMAT")
glmm.wald(disease ~ age + sex, data = pheno, kins = GRM, id = "id",
        family = binomial(link = "logit"), infile = infile, snps = snps)
}
infile <- system.file("extdata", "geno.txt", package = "GMMAT")
glmm.wald(disease ~ age + sex, data = pheno, kins = GRM, id = "id",
family = binomial(link = "logit"), infile = infile, snps = snps,
infile.nrow.skip = 5, infile.ncol.skip = 3, infile.ncol.print = 1:3,
infile.header.print = c("SNP", "Allele1", "Allele2"))
```

`glmmkin` *Fit generalized linear mixed model with known relationship matrices*

Description

Fit a generalized linear mixed model with a random intercept, or a random intercept and an optional random slope of time effect for longitudinal data. The covariance matrix of the random intercept is proportional to a known relationship matrix (e.g. kinship matrix in genetic association studies). Alternatively, it can be a variance components model with multiple random effects, and each component has a known relationship matrix.

Usage

```
glmmkin(fixed, data = parent.frame(), kins = NULL, id, random.slope = NULL,
groups = NULL, family = binomial(link = "logit"), method = "REML",
method.optim = "AI", maxiter = 500, tol = 1e-5, taumin = 1e-5,
taumax = 1e5, tauregion = 10, verbose = FALSE, ...)
```

Arguments

`fixed`	an object of class `formula` (or one that can be coerced to that class): a symbolic description of the fixed effects model to be fitted.
`data`	a data frame or list (or object coercible by `as.data.frame` to a data frame) containing the variables in the model.
`kins`	a known positive semi-definite relationship matrix (e.g. kinship matrix in genetic association studies) or a list of known positive semi-definite relationship matrices. The rownames and colnames of these matrices must at least include all samples as specified in the `id` column of the data frame `data`. If not provided, glmmkin will switch to the generalized linear model with no random effects (default = NULL).
`id`	a column in the data frame `data`, indicating the id of samples. When there are duplicates in `id`, the data is assumed to be longitudinal with repeated measures.
`random.slope`	an optional column indicating the random slope for time effect used in a mixed effects model for cross-sectional data with related individuals, and longitudinal data. It must be included in the names of `data`. There must be duplicates in `id` and `method.optim` must be "AI" (default = NULL).

groups	an optional categorical variable indicating the groups used in a heteroscedastic linear mixed model (allowing residual variances in different groups to be different). This variable must be included in the names of data, and family must be "gaussian" and method.optim must be "AI" (default = NULL).
family	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See family for details of family functions.)
method	method of fitting the generalized linear mixed model. Either "REML" or "ML" (default = "REML").
method.optim	optimization method of fitting the generalized linear mixed model. Either "AI", "Brent" or "Nelder-Mead" (default = "AI").
maxiter	a positive integer specifying the maximum number of iterations when fitting the generalized linear mixed model (default = 500).
tol	a positive number specifying tolerance, the difference threshold for parameter estimates below which iterations should be stopped (default = 1e-5).
taumin	the lower bound of search space for the variance component parameter τ (default = 1e-5), used when method.optim = "Brent". See Details.
taumax	the upper bound of search space for the variance component parameter τ (default = 1e5), used when method.optim = "Brent". See Details.
tauregion	the number of search intervals for the REML or ML estimate of the variance component parameter τ (default = 10), used when method.optim = "Brent". See Details.
verbose	a logical switch for printing detailed information (parameter estimates in each iteration) for testing and debugging purpose (default = FALSE).
...	additional arguments that could be passed to glm.

Details

Generalized linear mixed models (GLMM) are fitted using the penalized quasi-likelihood (PQL) method proposed by Breslow and Clayton (1993). Generally, fitting a GLMM is computationally expensive, and by default we use the Average Information REML algorithm (Gilmour, Thompson and Cullis, 1995; Yang et al., 2011) to fit the model. If only one relationship matrix is specified (kins is a matrix), iterations may be accelerated using the algorithm proposed by Zhou and Stephens (2012) for linear mixed models. An eigendecomposition is performed in each outer iteration and the estimate of the variance component parameter τ is obtained by maximizing the profiled log restricted likelihood (or likelihood) in a search space from taumin to taumax, equally divided into tauregion intervals on the log scale, using Brent's method (1973). If kins is a list of matrices and method = "Nelder-Mead", iterations are performed as a multi-dimensional maximization problem solved by Nelder and Mead's method (1965). It can be very slow, and we do not recommend using this method unless the likelihood function is badly behaved. Both Brent's method and Nelder and Mead's method are derivative-free. When the Average Information REML algorithm fails to converge, a warning message is given and the algorithm is default to derivative-free approaches: Brent's method if only one relationship matrix is specified, Nelder and Mead's method if more than one relationship matrix is specified.

For longitudinal data (with duplicated id), two types of models can be applied: random intercept only models, and random intercept and random slope models. The random intercept only model

is appropriate for analyzing repeated measures with no time trends, and observations for the same individual are assumed to be exchangeable. The random intercept and random slope model is appropriate for analyzing longitudinal data with individual-specific time trends (therefore, a random slope for time effect). Typically, the time effect should be included in the model as a fixed effect covariate as well. Covariances of the random intercept and the random slope are estimated.

For multiple phenotype analysis, `formula` recognized by `lm`, such as `cbind(y1, y2, y3) ~ x1 + x2`, can be used in `fixed` as fixed effects. For each matrix in `kins`, variance components corresponding to each phenotype, as well as their covariance components, will be estimated. Currently, `family` must be "gaussian" and `method.optim` must be "AI".

Value

`theta` a vector or a list of variance component parameter estimates. See below.

For cross-sectional data, if `kins` is not provided (unrelated individuals), `theta` is the dispersion parameter estimate from the generalized linear model; if `kins` is a matrix and `groups` is not provided, `theta` is a length 2 vector, with `theta[1]` being the dispersion parameter estimate and `theta[2]` being the variance component parameter estimate for `kins`; if `kins` is a list and `groups` is not provided, `theta` is a length `1 + length(kins)` vector, with `theta[1]` being the dispersion parameter estimate and `theta[2:(1 + length(kins))]` being the variance component parameter estimates, corresponding to the order of matrices in the list `kins`; if `kins` is a matrix and `groups` is provided (a heteroscedastic linear mixed model with `n.groups` residual variance groups), `theta` is a length `1 + n.groups` vector, with `theta[1:n.groups]` being the residual variance estimates for each group and `theta[1 + n.groups]` being the variance component parameter estimate for `kins`; if `kins` is a list and `groups` is provided (a heteroscedastic linear mixed model with `n.groups` residual variance groups), `theta` is a length `length(kins) + n.groups` vector, with `theta[1:n.groups]` being the residual variance estimates for each group and `theta[(1 + n.groups):(length(kins) + n.groups)]` being the variance component parameter estimates, corresponding to the order of matrices in the list `kins`.

For longitudinal data (with duplicated `id`) in a random intercept only model, if `kins` is not provided (unrelated individuals) and `groups` is not provided, `theta` is a length 2 vector, with `theta[1]` being the dispersion parameter estimate and `theta[2]` being the variance component parameter estimate for the random individual effects; if `kins` is a matrix and `groups` is not provided, `theta` is a length 3 vector, with `theta[1]` being the dispersion parameter estimate, `theta[2]` being the variance component parameter estimate for the random individual effects attributable to relatedness from `kins`, and `theta[3]` being the variance component parameter estimate for the random individual effects not attributable to relatedness from `kins`; if `kins` is a list and `groups` is not provided, `theta` is a length `2 + length(kins)` vector, with `theta[1]` being the dispersion parameter estimate, `theta[2:(1 + length(kins))]` being the variance component parameter estimates for the random individual effects attributable to relatedness from `kins`, corresponding to the order of matrices in the list `kins`, and `theta[2 + length(kins)]` being the variance component parameter estimate for the random individual effects not attributable to relatedness from `kins`; if `kins` is not provided (unrelated individuals) and `groups` is pro-

vided (a heteroscedastic linear mixed model with n.groups residual variance groups), theta is a length 1 + n.groups vector, with theta[1:n.groups] being the residual variance estimates for each group and theta[1 + n.groups] being the variance component parameter estimate for the random individual effects; if kins is a matrix and groups is provided (a heteroscedastic linear mixed model with n.groups residual variance groups), theta is a length 2 + n.groups vector, with theta[1:n.groups] being the residual variance estimates for each group, theta[1 + n.groups] being the variance component parameter estimate for the random individual effects attributable to relatedness from kins, and theta[2 + n.groups] being the variance component parameter estimate for the random individual effects not attributable to relatedness from kins; if kins is a list and groups is provided (a heteroscedastic linear mixed model with n.groups residual variance groups), theta is a length 1 + length(kins) + n.groups vector, with theta[1:n.groups] being the residual variance estimates for each group, theta[(1 + n.groups):(length(kins) + n.groups)] being the variance component parameter estimates for the random individual effects attributable to relatedness from kins, corresponding to the order of matrices in the list kins, and theta[1 + length(kins) + n.groups] being the variance component parameter estimate for the random individual effects not attributable to relatedness from kins.

For longitudinal data (with duplicated id) in a random intercept and random slope (for time effect) model, if kins is not provided (unrelated individuals) and groups is not provided, theta is a length 4 vector, with theta[1] being the dispersion parameter estimate, theta[2] being the variance component parameter estimate for the random individual effects of the intercept, theta[3] being the covariance estimate for the random individual effects of the intercept and the random individual effects of the time slope, and theta[4] being the variance component parameter estimate for the random individual effects of the time slope; if kins is a matrix and groups is not provided, theta is a length 7 vector, with theta[1] being the dispersion parameter estimate, theta[2] being the variance component parameter estimate for the random individual effects of the intercept attributable to relatedness from kins, theta[3] being the variance component parameter estimate for the random individual effects of the intercept not attributable to relatedness from kins, theta[4] being the covariance estimate for the random individual effects of the intercept and the random individual effects of the time slope attributable to relatedness from kins, theta[5] being the covariance estimate for the random individual effects of the intercept and the random individual effects of the time slope not attributable to relatedness from kins, theta[6] being the variance component parameter estimate for the random individual effects of the time slope attributable to relatedness from kins, and theta[7] being the variance component parameter estimate for the random individual effects of the time slope not attributable to relatedness from kins; if kins is a list and groups is not provided, theta is a length 4 + 3 * length(kins) vector, with theta[1] being the dispersion parameter estimate, theta[2:(1 + length(kins))] being the variance component parameter estimates for the random individual effects of the intercept attributable to relatedness from kins, corresponding to the order of matrices in the list kins, theta[2 + length(kins)] being the variance

component parameter estimate for the random individual effects of the intercept not attributable to relatedness from kins, theta[(3 + length(kins)):(2 + 2 * length(kins))] being the covariance estimates for the random individual effects of the intercept and the random individual effects of the time slope attributable to relatedness from kins, corresponding to the order of matrices in the list kins, theta[3 + 2 * length(kins)] being the covariance estimate for the random individual effects of the intercept and the random individual effects of the time slope not attributable to relatedness from kins, theta[(4 + 2 * length(kins)):(3 + 3 * length(kins))] being the variance component parameter estimates for the random individual effects of the time slope attributable to relatedness from kins, corresponding to the order of matrices in the list kins, theta[4 + 3 * length(kins)] being the variance component parameter estimate for the random individual effects of the time slope not attributable to relatedness from kins; if kins is not provided (unrelated individuals) and groups is provided (a heteroscedastic linear mixed model with n.groups residual variance groups), theta is a length 3 + n.groups vector, with theta[1:n.groups] being the residual variance estimates for each group, theta[1 + n.groups] being the variance component parameter estimate for the random individual effects of the intercept, theta[2 + n.groups] being the covariance estimate for the random individual effect of the intercept and the random individual effects of the time slope, and theta[3 + n.groups] being the variance component parameter estimate for the random individual effects of the time slope; if kins is a matrix and groups is provided (a heteroscedastic linear mixed model with n.groups residual variance groups), theta is a length 6 + n.groups vector, with theta[1:n.groups] being the residual variance estimates for each group, theta[1 + n.groups] being the variance component parameter estimate for the random individual effects of the intercept attributable to relatedness from kins, theta[2 + n.groups] being the variance component parameter estimate for the random individual effects of the intercept not attributable to relatedness from kins, theta[3 + n.groups] being the covariance estimate for the random individual effects of the intercept and the random individual effects of the time slope attributable to relatedness from kins, theta[4 + n.groups] being the covariance estimate for the random individual effects of the intercept and the random individual effects of the time slope not attributable to relatedness from kins, theta[5 + n.groups] being the variance component parameter estimate for the random individual effects of the time slope attributable to relatedness from kins, and theta[6 + n.groups] being the variance component parameter estimate for the random individual effects of the time slope not attributable to relatedness from kins; if kins is a list and groups is provided (a heteroscedastic linear mixed model with n.groups residual variance groups), theta is a length 3 + 3 * length(kins) + n.groups vector, with theta[1:n.groups] being the residual variance estimates for each group, theta[(1 + n.groups):(length(kins) + n.groups)] being the variance component parameter estimates for the random individual effects of the intercept attributable to relatedness from kins, corresponding to the order of matrices in the list kins, theta[1 + length(kins) + n.groups] being the variance component parameter estimate for the random individual effects of the intercept not attributable to relatedness from kins, theta[(2 + length(kins) + n.groups):(1 + 2 * length(kins) + n.groups)]

being the covariance estimates for the random individual effects of the intercept and the random individual effects of the time slope attributable to relatedness from kins, corresponding to the order of matrices in the list kins, theta[2 + 2 * length(kins) + n.groups] being the covariance estimate for the random individual effects of the intercept and the random individual effects of the time slope not attributable to relatedness from kins, theta[(3 + 2 * length(kins) + n.groups):(2 + 3 * length(kins) + n.groups)] being the variance component parameter estimates for the random individual effects of the time slope attributable to relatedness from kins, corresponding to the order of matrices in the list kins, and theta[3 + 3 * length(kins) + n.groups] being the variance component parameter estimate for the random individual effects of the time slope not attributable to relatedness from kins.

For multiple phenotype analysis, theta is a list of variance-covariance matrices. If kins is not provided (unrelated individuals), theta is an n.pheno by n.pheno variance-covariance matrix for the residuals of the multiple phenotypes from the linear model; if kins is a matrix and groups is not provided, theta is a length 2 list, with theta[[1]] being the variance-covariance matrix for the residuals and theta[[2]] being the variance-covariance matrix for kins; if kins is a list and groups is not provided, theta is a length 1 + length(kins) list, with theta[[1]] being the variance-covariance matrix for the residuals and theta[[2]] to theta[[1 + length(kins)]] being the variance-covariance matrices, corresponding to the order of matrices in the list kins; if kins is a matrix and groups is provided (a heteroscedastic linear mixed model with n.groups residual variance groups), theta is a length 1 + n.groups list, with theta[[1]] to theta[[n.groups]] being the variance-covariance matrices for the residuals in each group and theta[[1 + n.groups]] being the variance-covariance matrix for kins; if kins is a list and groups is provided (a heteroscedastic linear mixed model with n.groups residual variance groups), theta is a length length(kins) + n.groups list, with theta[[1]] to theta[[n.groups]] being the variance-covariance matrices for the residuals in each group and theta[[1 + n.groups]] to theta[[length(kins) + n.groups]] being the variance-covariance matrices, corresponding to the order of matrices in the list kins.

n.pheno — an integer indicating the number of phenotypes in multiple phenotype analysis (for single phenotype analysis, n.pheno = 1).

n.groups — an integer indicating the number of distinct residual variance groups in heteroscedastic linear mixed models (for other models, n.groups = 1).

coefficients — a vector or a matrix for the fixed effects parameter estimates (including the intercept).

linear.predictors — a vector or a matrix for the linear predictors.

fitted.values — a vector or a matrix for the fitted mean values on the original scale.

Y — a vector or a matrix for the final working vector.

X — model matrix for the fixed effects.

P — the projection matrix with dimensions equal to the sample size multiplied by n.pheno. Used in glmm.score and SMMAT for dense matrices.

`residuals` a vector or a matrix for the residuals on the original scale. NOT rescaled by the dispersion parameter.

`scaled.residuals` a vector or a matrix for the scaled residuals, calculated as the original residuals divided by the dispersion parameter (in heteroscedastic linear mixed models, corresponding residual variance estimates by each group).

`cov` covariance matrix for the fixed effects (including the intercept).

`Sigma_i` the inverse of the estimated covariance matrix for samples, with dimensions equal to the sample size multiplied by `n.pheno`. Used in `glmm.score` and `SMMAT` for sparse matrices.

`Sigma_iX` Sigma_i multiplied by X. Used in `glmm.score` and `SMMAT` for sparse matrices.

`converged` a logical indicator for convergence.

`call` the matched call.

`id_include` a vector indicating the `id` of rows in `data` with nonmissing outcome and covariates, thus are included in the model fit.

Author(s)

Han Chen, Matthew P. Conomos

References

Brent, R.P. (1973) "Chapter 4: An Algorithm with Guaranteed Convergence for Finding a Zero of a Function", Algorithms for Minimization without Derivatives, Englewood Cliffs, NJ: Prentice-Hall, ISBN 0-13-022335-2.

Breslow, N.E. and Clayton, D.G. (1993) Approximate Inference in Generalized Linear Mixed Models. Journal of the American Statistical Association 88, 9-25.

Chen, H., Wang, C., Conomos, M.P., Stilp, A.M., Li, Z., Sofer, T., Szpiro, A.A., Chen, W., Brehm, J.M., Celedón, J.C., Redline, S., Papanicolaou, G.J., Thornton, T.A., Laurie, C.C., Rice, K. and Lin, X. (2016) Control for population structure and relatedness for binary traits in genetic association studies via logistic mixed models. The American Journal of Human Genetics 98, 653-666.

Gilmour, A.R., Thompson, R. and Cullis, B.R. (1995) Average Information REML: An Efficient Algorithm for Variance Parameter Estimation in Linear Mixed Models. Biometrics 51, 1440-1450.

Nelder, J.A. and Mead, R. (1965) A simplex algorithm for function minimization. Computer Journal 7, 308-313.

Yang, J., Lee, S.H., Goddard, M.E. and Visscher, P.M. (2011) GCTA: A Tool for Genome-wide Complex Trait Analysis. The American Journal of Human Genetics 88, 76-82.

Zhou, X. and Stephens, M. (2012) Genome-wide efficient mixed-model analysis for association studies. Nature Genetics 44, 821-824.

Examples

```
data(example)
attach(example)
model0 <- glmmkin(disease ~ age + sex, data = pheno, kins = GRM, id = "id",
```

```
       family = binomial(link = "logit"))
model0$theta
model0$coefficients
model0$cov

model1 <- glmmkin(y.repeated ~ sex, data = pheno2, kins = GRM, id = "id",
       family = gaussian(link = "identity"))
model1$theta
model1$coefficients
model1$cov
model2 <- glmmkin(y.trend ~ sex + time, data = pheno2, kins = GRM, id = "id",
       random.slope = "time", family = gaussian(link = "identity"))
model2$theta
model2$coefficients
model2$cov
```

SMMAT *Variant Set Mixed Model Association Tests (SMMAT)*

Description

Variant Set Mixed Model Association Tests (SMMAT-B, SMMAT-S, SMMAT-O and SMMAT-E) for multiple user-defined test units and a null generalized linear mixed model. `SMMAT.prep` and `SMMAT.lowmem` are the two-step low-memory version of `SMMAT`. `SMMAT.lowmem` takes the returned R object from `SMMAT.prep` and uses less memory (if the returned R object from `SMMAT.prep` is saved to an R data file, the R session is terminated, and this R object is loaded into a new R session for running `SMMAT.lowmem`), especially when `group.file` contains only a subset of variants from `geno.file`.

Usage

```
SMMAT(null.obj, geno.file, group.file, group.file.sep = "\t",
meta.file.prefix = NULL, MAF.range = c(1e-7, 0.5),
MAF.weights.beta = c(1, 25), miss.cutoff = 1,
missing.method = "impute2mean", method = "davies",
tests = "E", rho = c(0, 0.1^2, 0.2^2, 0.3^2, 0.4^2,
0.5^2, 0.5, 1), use.minor.allele = FALSE,
auto.flip = FALSE, Garbage.Collection = FALSE,
is.dosage = FALSE, ncores = 1, verbose = FALSE)
SMMAT.prep(null.obj, geno.file, group.file, group.file.sep = "\t",
auto.flip = FALSE)
SMMAT.lowmem(SMMAT.prep.obj, geno.file = NULL, meta.file.prefix = NULL,
        MAF.range = c(1e-7, 0.5), MAF.weights.beta = c(1, 25),
miss.cutoff = 1, missing.method = "impute2mean",
method = "davies", tests = "E", rho = c(0, 0.1^2,
0.2^2, 0.3^2, 0.4^2, 0.5^2, 0.5, 1),
use.minor.allele = FALSE, Garbage.Collection = FALSE,
is.dosage = FALSE, ncores = 1, verbose = FALSE)
```

Arguments

null.obj	a class glmmkin or a class glmmkin.multi object, returned by fitting the null GLMM using glmmkin.
geno.file	the .gds file name or an object of class SeqVarGDSClass for the full genotypes. The sample.id in geno.file should overlap id_include in null.obj. It is recommended that sample.id in geno.file include the full samples (at least all samples as specified in id_include of null.obj). It is not necessary for the user to take a subset of geno.file before running the analysis. If geno.file is an object of class SeqVarGDSClass, the .gds file will be closed upon successful completion of the function.
group.file	a plain text file with 6 columns defining the test units. There should be no headers in the file, and the columns are group name, chromosome, position, reference allele, alternative allele and weight, respectively.
group.file.sep	the delimiter in group.file (default = "\t").
meta.file.prefix	prefix of intermediate files (.score.* and .var.*) required in a meta-analysis. If NULL, such intermediate files are not generated (default = NULL).
MAF.range	a numeric vector of length 2 defining the minimum and maximum minor allele frequencies of variants that should be included in the analysis (default = c(1e-7, 0.5)).
MAF.weights.beta	a numeric vector of length 2 defining the beta probability density function parameters on the minor allele frequencies. This internal minor allele frequency weight is multiplied by the external weight given by the group.file. To turn off internal minor allele frequency weight and only use the external weight given by the group.file, use c(1, 1) to assign flat weights (default = c(1, 25)).
miss.cutoff	the maximum missing rate allowed for a variant to be included (default = 1, including all variants).
missing.method	method of handling missing genotypes. Either "impute2mean" or "impute2zero" (default = "impute2mean").
method	a method to compute p-values for SKAT-type test statistics (default = "davies"). "davies" represents an exact method that computes a p-value by inverting the characteristic function of the mixture chisq distribution, with an accuracy of 1e-6. When "davies" p-value is less than 1e-5, it defaults to method "kuonen". "kuonen" represents a saddlepoint approximation method that computes the tail probabilities of the mixture chisq distribution. When "kuonen" fails to compute a p-value, it defaults to method "liu". "liu" is a moment-matching approximation method for the mixture chisq distribution.
tests	a character vector indicating which SMMAT tests should be performed ("B" for the burden test, "S" for SKAT, "O" for SKAT-O and "E" for the efficient hybrid test of the burden test and SKAT). The burden test and SKAT are automatically included when performing "O", and the burden test is automatically included when performing "E" (default = "E").
rho	a numeric vector defining the search grid used in SMMAT-O for SKAT-O (see the SKAT-O paper for details). Not used for SMMAT-B for the burden test,

SMMAT-S for SKAT or SMMAT-E for the efficient hybrid test of the burden test and SKAT (default = c(0, 0.1^2, 0.2^2, 0.3^2, 0.4^2, 0.5^2, 0.5, 1)).

`use.minor.allele`
: a logical switch for whether to use the minor allele (instead of the alt allele) as the coding allele (default = FALSE). It does not change SMMAT-S results, but SMMAT-B (as well as SMMAT-O and SMMAT-E) will be affected. Along with the MAF filter, this option is useful for combining rare mutations, assuming rare allele effects are in the same direction.

`auto.flip`
: a logical switch for whether to enable automatic allele flipping if a variant with alleles ref/alt is not found at a position, but a variant at the same position with alleles alt/ref is found (default = FALSE). Use with caution for whole genome sequence data, as both ref/alt and alt/ref variants at the same position are not uncommon, and they are likely two different variants, rather than allele flipping.

`Garbage.Collection`
: a logical switch for whether to enable garbage collection in each test (default = FALSE). Pay for memory efficiency with slower computation speed.

`is.dosage`
: a logical switch for whether imputed dosage should be used from `geno.file` (default = FALSE).

`ncores`
: a positive integer indicating the number of cores to be used in parallel computing (default = 1).

`verbose`
: a logical switch for whether a progress bar should be shown (default = FALSE).

`SMMAT.prep.obj`
: a class SMMAT.prep object, returned by `SMMAT.prep`.

Value

`SMMAT` and `SMMAT.lowmem` return a data frame with the following components:

`group`
: name of the test unit group.

`n.variants`
: number of variants in the test unit group that pass the missing rate and allele frequency filters.

`miss.min`
: minimum missing rate for variants in the test unit group.

`miss.mean`
: mean missing rate for variants in the test unit group.

`miss.max`
: maximum missing rate for variants in the test unit group.

`freq.min`
: minimum coding allele frequency for variants in the test unit group.

`freq.mean`
: mean coding allele frequency for variants in the test unit group.

`freq.max`
: maximum coding allele frequency for variants in the test unit group.

`B.score`
: burden test score statistic.

`B.var`
: variance of burden test score statistic.

`B.pval`
: burden test p-value.

`S.pval`
: SKAT p-value.

`O.pval`
: SKAT-O p-value.

`O.minp`
: minimum p-value in the SKAT-O search grid.

`O.minp.rho`
: rho value at the minimum p-value in the SKAT-O search grid.

E.pval	SMMAT efficient hybrid test of the burden test and SKAT p-value.

SMMAT.prep return a list with the following components:

null.obj	a class glmmkin or a class glmmkin.multi object from the null model, after pre-processing.
geno.file	the name of the .gds file for the full genotypes.
group.file	the name of the plain text file with 6 columns defining the test units.
group.file.sep	the delimiter in group.file.
auto.flip	a logical indicator showing whether automatic allele flipping is enabled in pre-processing if a variant with alleles ref/alt is not found at a position, but a variant at the same position with alleles alt/ref is found.
residuals	residuals from the null model, after pre-processing.
sample.id	sample.id from geno.file, after pre-processing.
group.info	group.info read from group.file, after pre-processing.
groups	unique groups in group.info, after pre-processing.
group.idx.start	a vector of the start variant index for each group, after pre-processing.
group.idx.end	a vector of the end variant index for each group, after pre-processing.

Author(s)

Han Chen

References

Wu, M.C., Lee, S., Cai, T., Li, Y., Boehnke, M., Lin, X. (2011) Rare-variant association testing for sequencing data with the sequence kernel association test. The American Journal of Human Genetics 89, 82-93.

Lee, S., Wu, M.C., Lin, X. (2012) Optimal tests for rare variant effects in sequencing association studies. Biostatistics 13, 762-775.

Sun, J., Zheng, Y., Hsu, L. (2013) A unified mixed-effects model for rare-variant association in sequencing studies. Genetic Epidemiology 37, 334-344.

Chen, H., Huffman, J.E., Brody, J.A., Wang, C., Lee, S., Li, Z., Gogarten, S.M., Sofer, T., Bielak, L.F., Bis, J.C., et al. (2019) Efficient variant set mixed model association tests for continuous and binary traits in large-scale whole-genome sequencing studies. The American Journal of Human Genetics 104, 260-274.

See Also

glmmkin, SMMAT.meta

Examples

```
if(requireNamespace("SeqArray", quietly = TRUE) && requireNamespace("SeqVarTools",
        quietly = TRUE)) {
data(example)
attach(example)
model0 <- glmmkin(disease ~ age + sex, data = pheno, kins = GRM, id = "id",
        family = binomial(link = "logit"))
geno.file <- system.file("extdata", "geno.gds", package = "GMMAT")
group.file <- system.file("extdata", "SetID.withweights.txt",
package = "GMMAT")
out <- SMMAT(model0, geno.file, group.file, MAF.range = c(0, 0.5),
        miss.cutoff = 1, method = "davies")
print(out)
}

## Not run:
obj <- SMMAT.prep(model0, geno.file, group.file)
save(obj, file = "SMMAT.prep.tmp.Rdata")
# quit R session
# open a new R session
obj <- get(load("SMMAT.prep.tmp.Rdata"))
out <- SMMAT.lowmem(obj, MAF.range = c(0, 0.5), miss.cutoff = 1,
        method = "davies")
print(out)
unlink("SMMAT.prep.tmp.Rdata")

## End(Not run)
```

`SMMAT.meta` *Meta-analysis for variant Set Mixed Model Association Tests (SMMAT)*

Description

Variant Set Mixed Model Association Tests (SMMAT-B, SMMAT-S, SMMAT-O and SMMAT-E) in the meta-analysis.

Usage

```
SMMAT.meta(meta.files.prefix, n.files = rep(1, length(meta.files.prefix)),
cohort.group.idx = NULL, group.file, group.file.sep = "\t",
MAF.range = c(1e-7, 0.5), MAF.weights.beta = c(1, 25),
miss.cutoff = 1, method = "davies", tests = "E", rho = c(0, 0.1^2,
0.2^2, 0.3^2, 0.4^2, 0.5^2, 0.5, 1), use.minor.allele = FALSE,
verbose = FALSE)
```

Arguments

`meta.files.prefix`
: a character vector for prefix of intermediate files (.score.* and .var.*) required in a meta-analysis. Each element represents the prefix of .score.* and .var.* from one cohort. The length of vector should be equal to the number of cohorts.

`n.files`
: an integer vector with the same length as meta.files.prefix, indicating how many sets of intermediate files (.score.* and .var.*) are expected from each cohort, usually as the result of multi-threading in creating the intermediate files (default = rep(1, length(meta.files.prefix))).

`cohort.group.idx`
: a vector with the same length as meta.files.prefix, indicating which cohorts should be grouped together in the meta-analysis assuming homogeneous genetic effects. For example, c("a","b","a","a","b") means cohorts 1, 3, 4 are assumed to have homogeneous genetic effects, and cohorts 2, 5 are in another group with homogeneous genetic effects (but possibly heterogeneous with group "a"). If NULL, all cohorts are in the same group (default = NULL).

`group.file`
: a plain text file with 6 columns defining the test units. There should be no headers in the file, and the columns are group name, chromosome, position, reference allele, alternative allele and weight, respectively.

`group.file.sep`
: the delimiter in group.file (default = "\t").

`MAF.range`
: a numeric vector of length 2 defining the minimum and maximum minor allele frequencies of variants that should be included in the analysis (default = c(1e-7, 0.5)). Filter applied to the combined samples.

`MAF.weights.beta`
: a numeric vector of length 2 defining the beta probability density function parameters on the minor allele frequencies. This internal minor allele frequency weight is multiplied by the external weight given by the group.file. To turn off internal minor allele frequency weight and only use the external weight given by the group.file, use c(1, 1) to assign flat weights (default = c(1, 25)). Applied to the combined samples.

`miss.cutoff`
: the maximum missing rate allowed for a variant to be included (default = 1, including all variants). Filter applied to the combined samples.

`method`
: a method to compute p-values for SKAT-type test statistics (default = "davies"). "davies" represents an exact method that computes a p-value by inverting the characteristic function of the mixture chisq distribution, with an accuracy of 1e-6. When "davies" p-value is less than 1e-5, it defaults to method "kuonen". "kuonen" represents a saddlepoint approximation method that computes the tail probabilities of the mixture chisq distribution. When "kuonen" fails to compute a p-value, it defaults to method "liu". "liu" is a moment-matching approximation method for the mixture chisq distribution.

`tests`
: a character vector indicating which SMMAT tests should be performed ("B" for the burden test, "S" for SKAT, "O" for SKAT-O and "E" for the efficient hybrid test of the burden test and SKAT). The burden test and SKAT are automatically included when performing "O", and the burden test is automatically included when performing "E" (default = "E").

`rho` a numeric vector defining the search grid used in SMMAT-O for SKAT-O (see the SKAT-O paper for details). Not used for SMMAT-B for the burden test, SMMAT-S for SKAT or SMMAT-E for the efficient hybrid test of the burden test and SKAT (default = c(0, 0.1^2, 0.2^2, 0.3^2, 0.4^2, 0.5^2, 0.5, 1)).

`use.minor.allele` a logical switch for whether to use the minor allele (instead of the alt allele) as the coding allele (default = FALSE). It does not change SMMAT-S results, but SMMAT-B (as well as SMMAT-O and SMMAT-E) will be affected. Along with the MAF filter, this option is useful for combining rare mutations, assuming rare allele effects are in the same direction. Use with caution, as major/minor alleles may flip in different cohorts. In that case, minor allele will be determined based on the allele frequency in the combined samples.

`verbose` a logical switch for whether a progress bar should be shown (default = FALSE).

Value

a data frame with the following components:

`group` name of the test unit group.

`n.variants` number of variants in the test unit group that pass the missing rate and allele frequency filters.

`B.score` burden test score statistic.

`B.var` variance of burden test score statistic.

`B.pval` burden test p-value.

`S.pval` SKAT p-value.

`O.pval` SKAT-O p-value.

`O.minp` minimum p-value in the SKAT-O search grid.

`O.minp.rho` rho value at the minimum p-value in the SKAT-O search grid.

`E.pval` SMMAT efficient hybrid test of the burden test and SKAT p-value.

Author(s)

Han Chen

References

Wu, M.C., Lee, S., Cai, T., Li, Y., Boehnke, M., Lin, X. (2011) Rare-variant association testing for sequencing data with the sequence kernel association test. The American Journal of Human Genetics 89, 82-93.

Lee, S., Wu, M.C., Lin, X. (2012) Optimal tests for rare variant effects in sequencing association studies. Biostatistics 13, 762-775.

Sun, J., Zheng, Y., Hsu, L. (2013) A unified mixed-effects model for rare-variant association in sequencing studies. Genetic Epidemiology 37, 334-344.

Chen, H., Huffman, J.E., Brody, J.A., Wang, C., Lee, S., Li, Z., Gogarten, S.M., Sofer, T., Bielak, L.F., Bis, J.C., et al. (2019) Efficient variant set mixed model association tests for continuous and binary traits in large-scale whole-genome sequencing studies. The American Journal of Human Genetics 104, 260-274.

See Also

glmmkin, SMMAT

Examples

```
if(requireNamespace("SeqArray", quietly = TRUE) && requireNamespace("SeqVarTools",
        quietly = TRUE)) {
data(example)
attach(example)
model0 <- glmmkin(disease ~ age + sex, data = pheno, kins = GRM, id = "id",
        family = binomial(link = "logit"))
geno.file <- system.file("extdata", "geno.gds", package = "GMMAT")
group.file <- system.file("extdata", "SetID.withweights.txt",
package = "GMMAT")
metafile <- tempfile()
out <- SMMAT(model0, geno.file, group.file, meta.file.prefix = metafile,
        MAF.range = c(0, 0.5), miss.cutoff = 1, method = "davies")
print(out)
out1 <- SMMAT.meta(metafile, group.file = group.file)
print(out1)
unlink(paste0(metafile, c(".score", ".var"), ".1"))
}
```

Index