Package 'RepeatABEL'

May 7, 2026

Title GWAS for Multiple Observations on Related Individuals

Version 2.0

Description Performs genome-wide association studies (GWAS) on individuals that are both related and have repeated measurements. For each Single Nucleotide Polymorphism (SNP), it computes score statistic based p-values for a linear mixed model including random polygenic effects and a random effect for repeated measurements. The computed p-values can be visualized in a Manhattan plot. For more details see Ronnegard et al. (2016) <doi:10.1111/2041-210X.12535> and for more examples see <https://github.com/larsronn/RepeatABEL_Tutorials>.

License GPL

Depends hglm, methods

Encoding UTF-8

RoxygenNote 7.3.3

Collate 'Compute_GRM2.R' 'Create_gwaa_data2.R' 'Create_gwaa_scan2.R' 'SmoothSNPmatrix.R' 'chromosome.R' 'constructV.R' 'estlambda2.R' 'snp.data.R' 'gwaa.data2.R' 'idnames.R' 'keep_gwaa_data.R' 'map.R' 'nids.R' 'plot.scan.gwaa2.R' 'preFitModel.R' 'rGLS.R' 'scan.gwaa2.R' 'simulate_PhenData.R' 'simulate_gendata.R' 'snpnames.R' 'summary_scan_gwaa2.R'

NeedsCompilation no

Author Lars Ronnegard [aut, cre]

Maintainer Lars Ronnegard <lars.ronnegard@slu.se>

Repository CRAN

Date/Publication 2026-01-06 11:00:15 UTC

Contents

`chromosome` *Extracts the chromosome numbers*

Description

Gets the chromosome numbers.

Usage

```
chromosome(genabel.data)
```

Arguments

`genabel.data`	A GenABEL-like data of class gwaa.data2.

Value

Returns an array of chromosome numbers

Author(s)

Lars Ronnegard

compute.GRM *Computes a Genetic Relationship Matrix from a GenABEL-like object*

Description

One method for GRM computations implemented.

Usage

```
compute.GRM(gen.data)
```

Arguments

gen.data	The GenABEL-like object.

Value

Returns a genomic relationship matrix.

Author(s)

Lars Ronnegard

constructV *Constructs the (co)variance matrix for y*

Description

Constructs the (co)variance matrix for y.

Usage

```
constructV(Z, RandC, ratio)
```

Arguments

Z	The incidence matrix for the random effects column binded with the Cholesky of the GRM
RandC	The number of columns in the two matrices combined in Z.
ratio	The ratios between random effect variances and the residual variance.

Value

Returns a (co)variance matrix of y.

Author(s)

Lars Ronnegard

Create_gwaa_data2 *Creates a gwaa.data2 object*

Description

Creates a gwaa.data2 object from input.

Usage

```
Create_gwaa_data2(genotypes, chromosome = NULL, map = NULL, phenotypes = NULL)
```

Arguments

genotypes	A matrix with genotype values coded as (0,1,2) or (-1,0,1)
chromosome	An array of characters for the chromosomes. Length equal to the number of SNPs.
map	An array with the order of the SNPs.
phenotypes	A data frame including columns with phenotypes and a column with ids, called "id"

Value

Returns a gwaa.data2-object.

Author(s)

Lars Ronnegard

Create_gwaa_scan2 *Creates a scan.gwaa2 object*

Description

Creates a scan.gwaa2 object from the rGLS output.

Usage

```
Create_gwaa_scan2(data, P1df, SNP.eff)
```

Arguments

data	A gwaa.data2 object
P1df	P-values computed from external analysis
SNP.eff	Estimated additive SNP effects

Value

Returns a scan.gwaa2 object.

Author(s)

Lars Ronnegard

`estlambda` *Function to estimate lambda*

Description

Estimates lambda from P-values. Most code copied from the archived GenABEL package

Usage

```
estlambda(data, plot = FALSE, method = "regression", filter = TRUE)
```

Arguments

`data`	An array of P-values
`plot`	Logical. TRUE to produce a plot
`method`	Either "regression" or "median".
`filter`	Logical. If TRUE the extreme P-values are not included in the estimate of lambda.

Value

Returns a list with estimate and standard error.

Author(s)

Lars Ronnegard

`gwaa.data2-class` *An S4 class to represent GWAS input data*

Description

An S4 class to represent GWAS input data

Slots

`phdata` Phenotype information including id

`gtdata` object of class `snp.data` with genotype information

idnames *Extracts the id names*

Description

Gets the idnames.

Usage

```
idnames(genabel.data)
```

Arguments

genabel.data	A GenABEL-like data of class gwaa.data2.

Value

Returns an array with the names of the individuals (as character).

Author(s)

Lars Ronnegard

keep_gwaa_data *A function to subset an gwaa.data2 object*

Description

Extracts a subset of the data.

Usage

```
keep_gwaa_data(genabel.data, indx.keep = NULL)
```

Arguments

genabel.data	A GenABEL-like data of class gwaa.data2.
indx.keep	Indeces to extract.

Author(s)

Lars Ronnegard

map *Extracts the map information*

Description

Gets the map.

Usage

```
map(genabel.data)
```

Arguments

genabel.data	A GenABEL-like data of class gwaa.data2.

Author(s)

Lars Ronnegard

nids *Extracts the number of ids*

Description

Gets nids.

Usage

```
nids(genabel.data)
```

Arguments

genabel.data	A GenABEL-like data of class gwaa.data2.

Value

Returns the number of individuals.

Author(s)

Lars Ronnegard

`plot.scan.gwaa2` *Function to plot P-values as a Manhattan plot*

Description

Creates a Manhattan plot

Usage

```
## S3 method for class 'scan.gwaa2'
plot(
  x,
  y,
  ...,
  ystart = 0,
  col = c("blue", "green"),
  sort = TRUE,
  ylim,
  main = NULL
)
```

Arguments

`x`	A scan.gwaa2 object created by the rGLS function
`y`	A parameter not used in the current version
`...`	Possible additional parameters (not used in the current version)
`ystart`	Lowest value on the y-axis
`col`	Default is c("blue","green")
`sort`	Logical. If TRUE the SNPs are sorted before plotting.
`ylim`	Limits of the y-axis
`main`	Plot title

Value

No return value, called for side effects

Author(s)

Lars Ronnegard

`preFitModel` *Fits a linear mixed model (without fixed SNP effects) and computes the fitted variance-covariance matrix for later use in the rGLS function.*

Description

Uses a GenABEL-like object and phenotype data as input. The model is fitted using the `hglm` function in the hglm package.

Usage

```
preFitModel(
  fixed = y ~ 1,
  random = ~1 | id,
  id.name = "id",
  genabel.data,
  phenotype.data,
  corStruc = NULL,
  GRM = NULL,
  Neighbor.Matrix = NULL,
  verbose = TRUE
)
```

Arguments

`fixed`	A formula including the response and fixed effects
`random`	A formula for the random effects
`id.name`	The column name of the IDs in phen.data
`genabel.data`	An GenABEL-like object including marker information. This object has one observation per individual.
`phenotype.data`	A data frame including the repeated observations and IDs.
`corStruc`	A list specifying the correlation structure for each random effect. The options are: `"Ind"` for iid random effects, `"GRM"` for a correlation structure given by a genetic relationship matrix, or `"CAR"` for a spatial correlation structure given by a Conditional Autoregressive model specified by a neighborhood matrix.
`GRM`	A genetic relationship matrix. If not specified whilst the `"GRM"` option is given for `corStruc` then the GRM is computed internally within the function.
`Neighbor.Matrix`	A neighborhood matrix having non-zero value for an element (i,j) where the observations i and j come from neighboring locations. The diagonal elements should be zero.
`verbose`	If TRUE the progress of the computations is printed.

Value

Returns a list including the fitted hglm object `fitted.hglm`, the variance-covariance matrix `V` and the ratios between estimated variance components for the random effects divided by the residual variance, `ratio`.

Author(s)

Lars Ronnegard

Examples

```
####### FIRST EXAMPLE USING GRM #############
set.seed(1234)
Gen.Data <- simulate_gendata(n=100, p=200)
Phen.Data <- simulate_PhenData(y ~ 1, genabel.data=Gen.Data,
                      n.obs=rep(4, nids(Gen.Data)), SNP.eff=2, SNP.nr=100, VC=c(1,1,1))
GWAS1 <- rGLS(y ~ 1, genabel.data = Gen.Data, phenotype.data = Phen.Data)
plot(GWAS1, main="")
summary(GWAS1)
#Summary for variance component estimation without SNP effects
summary(GWAS1@call$hglm)
#The same results can be computed using the preFitModel as follows
fixed = y ~ 1
Mod1 <- preFitModel(fixed, random=~1|id, genabel.data = Gen.Data,
                    phenotype.data = Phen.Data, corStruc=list( id=list("GRM","Ind") ))
GWAS1b <- rGLS(fixed, genabel.data = Gen.Data,
               phenotype.data = Phen.Data, V = Mod1$V)
plot(GWAS1b, main="Results using the preFitModel function")
```

`rGLS` *GWAS for Studies having Repeated Measurements on Related Individuals*

Description

It is used to perform genome-wide association studies on individuals that are both related and have repeated measurements. The function computes score statistic based p-values for a linear mixed model including random polygenic effects and a random effect for repeated measurements. A p-value is computed for each marker and the null hypothesis tested is a zero additive marker effect.

Usage

```
rGLS(
  formula.FixedEffects = y ~ 1,
  genabel.data,
  phenotype.data,
  id.name = "id",
```

```
  GRM = NULL,
  V = NULL,
  memory = 1e+08,
  verbose = TRUE
)
```

Arguments

formula.FixedEffects	Formula including the response variable and cofactors as fixed effects.
genabel.data	A GenABEL-like object including marker information. This object has one observation per individuals.
phenotype.data	A data frame including the repeated observations and IDs.
id.name	The column name of the IDs in phen.data
GRM	An optional genetic relationship matrix (GRM) can be included as input. Otherwise the GRM is computed within the function.
V	An optional (co)variance matrix can be included as input. Otherwise it is computed using the hglm function.
memory	Used to optimize computations. The maximum number of elements in a matrix that can be stored efficiently.
verbose	If TRUE the progress of the computations is printed.

Details

A generalized squares (GLS) is fitted for each marker given a (co)variance matrix V. The computations are made fast by transforming the GLS to an ordinary least-squares (OLS) problem using an eigen-decomposition of V. The OLS are computed using QR-factorization. If V is not specified then a model including random polygenic effects and permanent environmental effects is fitted (using the hglm package) to compute V. A GenABEL-like object (scan.gwaa2 class) is returned (including also the `hglm` results). Let e.g. GWAS1 be an object returned by the `rGLS` function. Then a Manhattan plot can be produced by calling `plot(GWAS1)` and the top SNPs using `summary(GWAS1)`.
The results from the fitted linear mixed model without any SNP effect included are produced by calling `summary(GWAS1@call$hglm)`.

Value

Returns a gwaa.scan2-object.

Author(s)

Lars Ronnegard

Examples

```
set.seed(1234)
Gen.Data <- simulate_gendata(n=100, p=200)
Phen.Data <- simulate_PhenData(y ~ 1, genabel.data=Gen.Data,
                      n.obs=rep(4, nids(Gen.Data)), SNP.eff=2, SNP.nr=100, VC=c(1,1,1))
```

```
GWAS1 <- rGLS(y ~ 1, genabel.data = Gen.Data, phenotype.data = Phen.Data)
plot(GWAS1, main="")
summary(GWAS1)
#Summary for variance component estimation without SNP effects
summary(GWAS1@call$hglm)
```

---

scan.gwaa2-class *An S4 class to represent SNP data*

Description

An S4 class to represent SNP data

Slots

`results` The results from the rGLS function as a data.frame

`lambda` Computed inflation factor as list

`idnames` Idnames as character

`map` SNP order as numeric

`chromosome` The chromosome name for each SNP as numeric

`call` The call made by rGLS as call

`family` The assumed distribution of the outcome. Only "gaussian" allowed.

---

simulate_gendata *Function to simulate genotype data for the RepeatABEL package.*

Description

The function simulates n individuals and p SNPs, with linkage disequilibrium (LD) given by the LD-parameter

Usage

```
simulate_gendata(n = 100, p = 1000, LD = 0.9, n.chrom = 1)
```

Arguments

`n`	Number of individuals.
`p`	Number of SNPs.
`LD`	An LD-parameter. LD=1 gives complete LD and LD=0 no LD.
`n.chrom`	The size of a simulated SNP.effect.

Value

Returns a gwaa.data2 object.

Author(s)

Lars Ronnegard

Examples

```
set.seed(1234)
Gen.Data <- simulate_gendata(n=100, p=200)
```

`simulate_PhenData` *Simulation function for the RepeatABEL package.*

Description

The function takes a GenABEL-like object (class gwaa.data2) as input and generates simulated phenotypic values for related individuals having repeated obserevations.

Usage

```
simulate_PhenData(
  formula.FixedEffects = y ~ 1,
  genabel.data,
  n.obs,
  SNP.eff = NULL,
  SNP.nr = NULL,
  beta = NULL,
  VC = c(1, 1, 1),
  GRM = NULL,
  sim.gamma = FALSE
)
```

Arguments

`formula.FixedEffects`	A formula including the name of the simulated variable as response, and cofactors as fixed effects.
`genabel.data`	A GenABEL-like object of class gwaa.data2.
`n.obs`	A vector including the number of observations per individual. The length of n.obs must be equal to the number if individuals in genabel.data.
`SNP.eff`	The size of a simulated SNP.effect.
`SNP.nr`	The SNP genotype that the SNP effect is simulated on. SNP.nr=i is the i:th SNP.

beta	The simulated fixed effects. Must be equal to the number of cofactors simulated (including the intercept term).
VC	A vector of length 3 including the simulated variances of the polygenic effect, permanent environmental effect and residuals, respectively.
GRM	An optional input where the Genetic Relationship Matrix can be given. Otherwise it is computed using the GenABEL package.
sim.gamma	A logical parameter specifying whether the residuals shuld be simulated from a gamma distribution or not. If specified as TRUE then residuals are drawn from a gamma distribution with variance equal to the residual variance specified in VC[3]

Value

Returns a data frame including the simulated phenotypic values, cofactors and IDs.

Author(s)

Lars Ronnegard

Examples

```
#Simulate 4 observations per individual
set.seed(1234)
Gen.Data <- simulate_gendata(n=100, p=200)
Phen.Data <- simulate_PhenData(y ~ 1, genabel.data=Gen.Data,
                        n.obs=rep(4, nids(Gen.Data)), SNP.eff=2, SNP.nr=100, VC=c(1,1,1))
GWAS1 <- rGLS(y ~ 1, genabel.data = Gen.Data, phenotype.data = Phen.Data)
plot(GWAS1, main="Simulated Data Results")
```

SmoothSNPmatrix *Imputes column means to missing genotypes*

Description

Imputes column means to missing genotypes.

Usage

```
SmoothSNPmatrix(SNP)
```

Arguments

SNP	A matrix including SNP coding.

Author(s)

Lars Ronnegard

snp.data *An S4 class to represent SNP data*

Description

An S4 class to represent SNP data

Slots

`nids` The number of ids as numeric

`idnames` The idnames as character

`nsps` The number of SNPs as numeric

`snpnames` The SNP names as character

`map` The order of the SNPS as numeric

`chromosome` The chromosome names for each SNP as numeric

`gtps` The matrix with SNP coding

snpnames *Extracts the snpnames*

Description

Gets the SNP names.

Usage

```
snpnames(genabel.data)
```

Arguments

`genabel.data` A GenABEL-like data of class gwaa.data2.

Author(s)

Lars Ronnegard

`summary.scan.gwaa2` *Summary function for the rGLS output*

Description

Creates a Manhattan plot using a slimmed version of the summary.scan.gwaa() function in the GenABEL package

Usage

```
## S3 method for class 'scan.gwaa2'
summary(object, ...)
```

Arguments

`object`	A scan.gwaa2 object created by the rGLS function
`...`	Possible additional parameters (not used in the current version)

Value

Returns a data frame with estimated SNP effects, standard errors, test-statistic values, p-values, and corrected p-values.

Author(s)

Lars Ronnegard

Index