Package 'wISAM'

October 12, 2022

Type Package

Title Weighted Inbred Strain Association Mapping

Version 0.2.8

Author Robert W. Cort <rcorty@gmail.com>

Maintainer Robert W. Cort <rcorty@gmail.com>

Description In the course of a genome-wide association study, the situation often arises that some phenotypes are known with greater precision than others.
It could be that some individuals are known to harbor more micro-environmental variance than others.
In the case of inbred strains of model organisms, it could be the case that more organisms were observed from some strains than others, so the strains with more organisms have better-estimated means.
Package 'wISAM' handles this situation by allowing for weighting of each observation according to residual variance.
Specifically, the 'weight' parameter to the function conduct_scan() takes the precision of each observation (one over the variance).

License GPL-3

Encoding UTF-8

Depends R (>= 3.0.0)

LazyData true

Suggests testthat

RoxygenNote 6.0.1

Imports methods, MASS

NeedsCompilation no

Repository CRAN

Date/Publication 2018-06-04 08:12:54 UTC

R topics documented:

`check_eigen_decomposition`
check_eigen_decomposition

Description

Grabbed from MASS. Useful to sparsify matrices when some eigenvalues are essentially zero.

Usage

```
check_eigen_decomposition(e, tol = 1e-06)
```

Arguments

`e`	the eigen decomposition to check
`tol`	the threshold below which a number is said to be effectively zero, defaults to 1e-6

Value

The eigen values with any values with absolute value less than tol zeroed.

covariate_mat *Example covariate matrix*

Description

Example covariate matrix

Usage

```
covariate_mat
```

Format

A matrix with 200 rows and 4 variables

GenomeScan-class *GenomeScan*

Description

A Reference Class implementing a Genome Scan

Fields

`Options` these are options

`Data` Things the user inputs. They have interpretable meaning and define the GenomeScan. Currently: y, X, G, K, weights (inverse variances), and variances.

`Intermediates_per_scan` Things the GenomeScan will compute once per scan. They are mathetmatical tools that can't really be interpreted. Currently: L, eigen_L, and LL_null.

`Intermediates_per_locus` Things the GenomeScan will compute once per locus. They are mathematical tools that can't really be interpreted. Currently: XG

`Intermediates_per_fit` Things the GenomeScan will compute many times per locus (once per trial fit on that locus). These are interpretable but rapidly changing and not guaranteed to be finalized or optimal. Currently: M, LDV, and h2

`Results` The results of the GenomeScan. Currently: The h2 that maximizes the LL at each locus and the LR as compared with the no-locus (null) model.

GenomeScan_calc_multiplier_eigen
calc_multiplier_eigen

Description

Compute a multiplier (aka rotation) matrix. Details in in h2lmm_math_RWC.Rmd.

Value

an object of class GenomeScan

GenomeScan_conduct_scan
conduct_scan

Description

Conducts the GenomeScan.

Value

an object of class GenomeScan

Note

TODO: allow user to specify subset of chromosomes or loci

Examples

```
library(wISAM)

wgs <- GenomeScan$new(y = phenotype,
                      X = covariate_mat,
                      G = locus_list,
                      K = kinship_mat,
                      w = 1/se_mean_per_strain)

result <- wgs$prep_scan()$conduct_scan()
```

GenomeScan_fit_locus *fit_locus*

Description

Fit one locus of a GenomeScan. Should not typically be called by a user.

Value

an object of class GenomeScan

GenomeScan_fit_locus_given_h2 *fit_locus_given_h2*

Description

Fit one locus at a specified value of h2. Should not typically be called by a user.

Value

an object of class GenomeScan

GenomeScan_initialize *initialize*

Description

Initialize a GenomeScan

Arguments

y	vector of length n - the phenotype of each of n genomes (individuals or strains)
X	matrix of dimension n-by-c - the covariate value of each individual for c covariates
G	a list where each element is of length n - the genotype of each individual at p loci
K	matrix of dimension n-by-n - the covariance of the phenotype
w	matrix of dimension n-by-n - the inverse variance of the phenotype

Value

an object of class GenomeScan

Examples

```
library(wISAM)

wgs <- GenomeScan$new(y = phenotype,
                      X = covariate_mat,
                      G = locus_list,
                      K = kinship_mat,
                      w = 1/se_mean_per_strain)
```

`GenomeScan_prep_scan` *prep_scan*

Description

Prepare a GenomeScan for running. Does all the computations that need to be done exactly once per genome scan.

Value

an object of class GenomeScan

Examples

```
library(wISAM)

wgs <- GenomeScan$new(y = phenotype,
                      X = covariate_mat,
                      G = locus_list,
                      K = kinship_mat,
                      w = 1/se_mean_per_strain)

result <- wgs$prep_scan()
```

`kinship_mat` *Example kinship matrix*

Description

Example kinship matrix

Usage

```
kinship_mat
```

Format

A matrix with 200 rows and 200 columns

locus_list *Example locus list*

Description

Example locus list

Usage

```
locus_list
```

Format

A list of matrices, where each matrix has 200 rows and 1 column

phenotype *Example phenotype*

Description

Example phenotype

Usage

```
phenotype
```

Format

A vector of length 200

se_mean_per_strain *Example standard error of the mean per strain*

Description

Example standard error of the mean per strain

Usage

```
se_mean_per_strain
```

Format

A vector of length 200

Index