Package ‘milorGWAS’

March 25, 2020

Type Package

Title Mixed Logistic Regression for Genome-Wide Analysis Studies (GWAS)

Version 0.3

Date 2020-03-13

Encoding UTF-8

Author Hervé Perdry and Jacqueline Milet

Maintainer Hervé Perdry <herve.perdry@u-psud.fr>

Description Fast approximate methods for mixed logistic regression in genome-wide analysis studies (GWAS). Milet et al 2020 <doi:10.1101/2020.01.17.910109>.

License GPL-3

Imports Rcpp (>= 1.0.2)

Depends gaston (>= 1.5.6)

LinkingTo Rcpp, RcppEigen, gaston

Suggests knitr, rmarkdown, png

VignetteBuilder knitr

NeedsCompilation yes

RoxygenNote 7.0.2

Repository CRAN

Date/Publication 2020-03-25 15:30:08 UTC

R topics documented:

`association.test.logistic` *Mixed logistic regression for GWAS*

Description

Mixed logistic regression for GWAS

Usage

```
association.test.logistic(
  x,
  Y = x@ped$pheno,
  X = matrix(1, nrow(x)),
  K,
  beg = 1,
  end = ncol(x),
  algorithm = c(”offset”, ”amle”),
  eigenK,
  p = 0,
  ...
)
```

Arguments

`x`	a bedmatrix
`Y`	phenotype vector. Default is column `pheno` of `x@ped`
`X`	A matrix of covariates (defaults to a column of ones for the intercept)
`K`	A genetic relationship matrix (or a list of such matrices)
`beg`	Index of the first SNP tested for association
`end`	Index of the last SNP tested for association
`algorithm`	Algorithm to use
`eigenK`	eigen decomposition of K (only if `p` > 0)
`p`	Number of principal components to include in the model
`...`	Additional parameter for `gaston::logistic.mm.aireml`

Details

Tests the association between the phenotype and requested SNPs in `x`. The phenotype `Y` is a binary trait. A Wald test is performed using an approximate method defined by the parameter `algorithm`. All other arguments are as in `gaston::association.test`.

Value

A data frame giving for each SNP the association statistics.

See Also

association.test

Examples

```
data(TTN)
x <- as.bed.matrix(TTN.gen, TTN.fam, TTN.bim)
## Simulation data ##
set.seed(1)
# some covariables
X <- cbind(1, runif(nrow(x)))
# A random GRM
ran <- random.pm( nrow(x))
# random effects (tau = 1)
omega <- lmm.simu(1, 0, eigenK=ran$eigen)$omega
# linear term of the model
lin <- X %*% c(0.1,-0.2) + omega
# vector of probabilitues
pi <- 1/(1+exp( -lin ))
# vector of binary phenotypes
y <- rbinom(nrow(x), 1, pi)
# testing association with 1) the score test, 2) the offset algorithm, 3) the 'amle' algorithm
a1 <- association.test(x, y, X, K = ran$K, method = "lmm", response = "bin")
a2 <- association.test.logistic(x, y, X, K = ran$K, algorithm = "offset")
a3 <- association.test.logistic(x, y, X, K = ran$K, algorithm = "amle")
```

---

qqplot.pvalues *Stratified QQ-plot of p-values*

Description

Draws a QQ plot of p-values

Usage

```
qqplot.pvalues(
  p,
  snp.cat,
  col.cat,
  col.abline = "red",
  CB = TRUE,
  col.CB = "gray80",
  CB.level = 0.95,
  thinning = TRUE,
  ...
)
```

Arguments

`p`	vector of p-values, or a data.frame with a column named `p`
`snp.cat`	(optional) A factor giving the SNP categories.
`col.cat`	(optional) A vector of colors used to plot the SNP categories.
`col.abline`	Color of the line of slope 1. Set to `NA` to suppress.
`CB`	`Logical`. If `TRUE`, a confidence band is included in the plot.
`col.CB`	The color of the confidence band.
`CB.level`	The level of the confidence band.
`thinning`	`Logical`. If `TRUE`, not all points are displayed.
`...`	Graphical parameters to be passed to `plot` and `points`

Details

This function draws a QQ plot of p-values, stratified by categories. If the parameter `snp.cat` is missing, the function falls back on `gaston::qqplot.pvalues`.

See Also

`SNP.category`, `qqplot.pvalues` (in gaston)

Examples

```
# a random vector of categories
ca <- sample(c("A","B","C"), 1e6, TRUE, c(0.05, 0.9, 0.05))
# a vector of p-values, with different distribution depending on the strata
p <- runif(1e6)**ifelse(ca == "A", .8, ifelse(ca == "B", 1, 1.2))
qqplot.pvalues(p, ca)
```

`SNP.category` *SNP.category*

Description

SNP.category

Usage

```
SNP.category(bed, Z, threshold = 0.8)
```

Arguments

`bed`	A bed matrix
`Z`	A vector of length `nrow(bed)`
`threshold`	Variance thresholds

Details

This function determines a SNP Category from a covariable Z, which can be for example an indicator variable for a population strata, or the first genomic principal component.

See Also

qqplot.pvalues

Examples

```
# a random vector of categories
ca <- sample(c("A","B","C"), 1e6, TRUE, c(0.05, 0.9, 0.05))
# a vector of p-values, with different distribution depending on the strata
p <- runif(1e6)**ifelse(ca == "A", .8, ifelse(ca == "B", 1, 1.2))
qqplot.pvalues(p, ca)
```

Index