Package ‘iGasso’

March 12, 2024

Type Package

Encoding UTF-8

Title Statistical Tests and Utilities for Genetic Association

Version 1.6.1

Date 2024-3-11

Author Dr. Kai Wang

Maintainer Kai Wang <kai-wang@uiowa.edu>

Depends lattice, CompQuadForm, MASS, MBESS

Description A collection of statistical tests for genetic association studies and summary data based Mendelian randomization.

License GPL (>= 2)

LazyLoad yes

RoxygenNote 7.3.1

NeedsCompilation no

Repository CRAN

Date/Publication 2024-03-12 00:40:02 UTC

R topics documented:

`iGasso-package` *Statistical Tests and utilities for Genetic Association*

Description

`iGasso` is a collection of statistical tests developed by our group for genetic association studies. So far it contains functions for rare variants association, for association with multiple phenotypes, for linear mixed model analysis, and for model-free association analysis. There is also a function for genome plot. It will keep growing as more tests are developed. Use `?iGasso` to see an introduction.

Details

Package:	iGasso
Type:	Package
Version:	1.6
Date:	2023-09-16
License:	GPL (>=2)
LazyLoad:	yes

Author(s)

Kai Wang <`kai-wang@uiowa.edu`>

References

Anscombe F.J. (1948) The transformation of Poisson, binomial and negative-binomial data. *Biometrika* **35(3/4)**, 246–254.

Chanter, D. O. (1975). Modifications of the angular transformation. *Journal of the Royal Statistical Society. Series B (Applied Statistics)*, **24 (3)**, 354–359.

Freeman, M. F., Tukey, J. W. (1950) Transformations related to the angular and the square root. *The Annals of Mathematical Statistics* **21(4)**, 607–611.

Wang, K. (2012) An application of the proportional odds model to genetic association studies. Submitted.

Wang K. (2012) Statistical tests of genetic association for case-control study designs. *Biostatistics*. Accepted. PMID: 22389176

Wang, K., Fingert, J. (2012) Statistical tests for detecting rare variants using variance-stabilizing transformations. *Annals of Human Genetics*. Accepted.

Zar, J. H. (1999) *Biostatistical Analysis, 4th ed.*, New Jersey:Prentice-Hall, Inc.

Examples

```
y = rnorm(100)
chr = c(rep(1, 20), rep(3, 20), rep(10, 20), rep(19, 30), rep("X", 10))
pos = c(1:20, 1:20, 1:20, 1:30, 1:10)
mydata = data.frame(y=y, chr=chr, pos=pos)
genome.plot(mydata, sig.line=c(1, -1), ylab="T Statistic")

G = rbind(c(14, 999), c(3, 1081))
VSTF.test(G)

G = rbind(c(161, 474, 489), c(231, 444, 380))
MFree.test(G)

G = matrix(sample(c(0,1,2), 200, replace=TRUE), ncol=10)
y = rnorm(10)
X = matrix(rnorm(10), ncol=1)
```

`genome.plot` *Genome-wide Plot of a Variable*

Description

`genome.plot` plots the value of a variable across the genome.

Usage

```
genome.plot(
  mydata,
  style = 1,
  type = "h",
  sig.line = c(4, -4),
  sig.color = c("red", "red"),
  ...
)
```

Arguments

`mydata` a data frame containing three variables: `y` (numeric, the value of the variable to be plotted), `chr` (character, chromosome label), and `pos` (numeric, position, for instance, in base pair or centi-Morgan). Examples of `y` include -log10 of p-values and test statistic values.

`style` either 1 (default) or 2.

`type` a generic graphic parameter. Recommended values are `"h"` (default) and `"b"`.

`sig.line` vertical locations of significance lines.

`sig.color` colors of significance lines.

`...` other parameters to be passed to function `xyplot` in the `lattice` package.

Details

This function makes use of the function `xyplot` from package `lattice`.

Author(s)

Kai Wang <`kai-wang@uiowa.edu`>

Examples

```
y = rnorm(100)
chr = c(rep(1, 20), rep(3, 20), rep(10, 20), rep(19, 30), rep("X", 10))
pos = c(1:20, 1:20, 1:20, 1:30, 1:10)
mydata = data.frame(y=y, chr=chr, pos=pos)
mydata2 = data.frame(y=y^2, chr=chr, pos=pos)

genome.plot(mydata, sig.line=c(1, -1), ylab="T Statistic")
genome.plot(mydata, sig.line=c(1, -1), ylab="T Statistic", type="b")
genome.plot(mydata2, sig.line=c(2), ylab="y squared")
genome.plot(mydata, style=2, sig.line=c(1, -1), ylab="T Statistic")
genome.plot(mydata, style=2, sig.line=c(1, -1), ylab="T Statistic", type="b")
```

`h2_snp` *An exact method for SNP-heritability estimation using GWAS summary statistics*

Description

`h2_snp` calculates heritability explained by a set of SNPs

Usage

```
h2_snp(beta, SE, N, R, alpha)
```

Arguments

`beta` a vector of beta coefficients for a set of SNPs. These coefficients are from a GWAS.

`SE` a vector of the standard errors of the beta coefficients.

`N` a vector of sample sizes used by the GWAS at these SNPs.

`R` LD matrix for these SNPs.

`alpha` $1 - \alpha$ is the confidence level of the confidence interval.

Value

A list containing the following components:

* MLE of the heritability.

* umvu (uniformly minimum variance unbiased) estimator of the heritability.

* interval estimate for the heritability.

Author(s)

Kai Wang <`kai-wang@uiowa.edu`>

References

Wang, K. (2023) An exact method for SNP-heritability estimation using GWAS summary statistics without heritability modeling. *submitted*

Examples

```
beta = c(0.225269, 0.221270, 0.162635, 0.261669, 0.150887,
         0.214515, 0.170296, 0.204454, 0.254811, 0.213803)
SE = c(0.033244, 0.032551, 0.032171, 0.031042, 0.032815,
       0.031908, 0.031717, 0.032023, 0.031907, 0.032291)
N = 10000
R = diag(1, 10)
alpha = 0.05
h2_snp(beta, SE, N, R, alpha)
```

`KAT.coin` *Conditional Inference for the Kernel Association Test (KAT)*

Description

Computes the asymptotic and the approximate conditional p-values for the kernel association test

Usage

```
KAT.coin(y, G, X = NULL, out_type = "D", distribution = "asymptotic", B = 1000)
```

Arguments

`y` a vector of phenotype on n subjects.

`G` an $n \times m$ matrix of SNP genotypes of n study subjects at m loci.

`X` a matrix of covariates. It has n rows.

`out_type` an indicator of the outcome type. `"C"` for the continuous outcome and `"D"` for the dichotomous outcome.

`distribution` a character, the conditional null distribution of the test statistic can be approximated by its asymptotic distribution ("asymptotic", default) or via Monte Carlo resampling ("approximate"), as in package `coin`.

`B` the number of permutations if `distribution = "approximate"`.

Details

The asymptotic conditional null distribution is obtained using results in Strasser and Weber (1999). The p-value based on this distribution is computed using Davies' method.

Value

A list with class "htest" containing the following components:

* statistic the value of the kernel association test statistic.

* parameter sample size and the number of SNPs.

* p.value the p-value based on the asymptotic or the approximate conditional null distribution.

* method a character string indicting the test performed.

* data.name a character string giving the name of the data.

Author(s)

Kai Wang <kai-wang@uiowa.edu>

References

Strasser, H. and Weber, C. (1999) On the asymptotic theory of permutation statistics. *Mathematical Methods of Statistics*. 8(2):220-250.

Wang, K. (2017) Conditional Inference for the Kernel Association Test. Bioinformatics 33 (23), 3733-3739.

Examples

```
n=1000
y = c(rep(1, n/2), rep(0, n/2))
maf = seq(0.05, 0.5, 0.05)
g = NULL
for (j in 1:10){
   geno.freq = c(maf[j]^2, 2*maf[j]*(1-maf[j]), (1-maf[j])^2)
   g = cbind(g, sample(c(0,1,2), n, replace=TRUE, prob=geno.freq))
   }
KAT.coin(y, g, X=NULL, out_type="D", B=1000)
```

MFree.test *Model-free Association Tests*

Description

MFree.test performs tests on association between an SNP and case-control status. It tests whether the frequencies of an allele are the same between cases and controls. It does not require specification of an inheritance model.

Usage

```
MFree.test(G, method = "score")
```

Arguments

G	a 2x3 two-dimensional contingency table in matrix form. The first row is for cases and the second one for controls. In each row, the entries are the number of subjects carrying 0, 1, and 2 copies of the reference allele, respective.
method	a character string indicating the test statistic to use. One of "score" (default), "Wald", and "LRT".

Details

Each test is named after the author(s) of the corresponding publication.

Value

A list with class "test" containing the following components: * statistic the value of the test statistic. * p.value the p-value for the test computed from a chi-square distribution with 1 df. * method a character string indicting the test performed. * data.name a character string giving the name of the data.

Author(s)

Kai Wang <kai-wang@uiowa.edu>

References

Wang K. (2012) Statistical tests of genetic association for case-control study designs. *Biostatistics*. 13(4):724-33. PMID: 22389176

Examples

```
G = rbind(c(161, 474, 489), c(231, 444, 380))
MFree.test(G)
MFree.test(G, method = "Wald")
MFree.test(G, method = "LRT")
```

MR_het_test *Test of Heterogeneity in MR using Principal Components*

Description

MR_het_test performs tests of heterogeneity in MR.

Usage

```
MR_het_test(x.b, y.b, x.se, y.se, b0, k = NULL, cum.prop = 0.8)
```

Arguments

`x.b` a vector of the estimated regression coefficients from the SNP-exposure GWAS

`y.b` a vector of the estimated regression coefficients from the SNP-outcome GWAS

`x.se` a vector of SEs for `x.b`

`y.se` a vector of SEs for `y.b`

`b0` a value used for the common effect size. It is used for the weighting matrix

`k` the number of principal components used. It is used by the $\tilde{Q}(b_0)$ statistic. The default is NULL

`cum.prop` threshold for selecting `k`. It is void if `k` is specified. The default is 0.8, i.e., the proportion of variance explained by the top `k` principal components is 0.8

Value

A list containing the following components:

* $P_{\text{min}}(b_0)$ statistic and its P-value.

* $\tilde{Q}_{\text{min}}(b_0)$ statistic, its degrees of freedom, and its P-value.

Author(s)

Kai Wang <`kai-wang@uiowa.edu`>

References

Wang, K, Alberding, Steven Y. (2024) Powerful test of heterogeneity in two-sample summary-data Mendelian randomization. Submitted.

Examples

```
p = 10
b = 0.5
gamma.true = runif(p, 0.34, 1.1)
x.se = runif(p, 0.06, 0.1)
y.se = runif(p, 0.015, 0.11)
x.b = rnorm(p, gamma.true, x.se)
y.b = rnorm(p, b*gamma.true, y.se)
b0 = 0.4

MR_het_test(x.b, y.b, x.se, y.se, b0)
```

SKATplus *A Gene- or Pathway-Based Test of Association* SKATlus *provides enhanced power over SKAT by properly estimating the null distribution of SKAT.*

Description

This version uses only subjects with lower phenotypic values for estimating the null distribution. That is, the "controls" are those of lower phenotypic values. When "controls" are of higher phenotypic values, change the sign of the phenotypic values in order to use this function.

Usage

```
SKATplus(
  y,
  G,
  X = NULL,
  out_type = "D",
  tau = NULL,
  permutation = FALSE,
  B = 1000
)
```

Arguments

y	a vector of phenotype on n subjects.
G	an $n \times m$ matrix of SNP genotypes of n study subjects at m loci.
X	a matrix of covariates. It has n rows.
out_type	an indicator of the outcome type. "C" for the continuous outcome and "D" for the dichotomous outcome.
tau	proportion of selected subjects used for SKATplus.
permutation	an indicator. Use permutation for p-value or not.
B	the number of permutations if permutation=TRUE

Value

A list with class "htest" containing the following components: * statistic the value of the test statistic, which is the same as SKAT statistic. * parameter sample size and the number of SNPs * p.value the p-value for SKATplus computed using Davies' method. * method a character string indicting the test performed. * data.name a character string giving the name of the data.

Author(s)

Kai Wang <kai-wang@uiowa.edu>

References

Wang, K. (2016) Boosting the power of the sequence kernel association test (SKAT) almost surely by properly estimating its null distribution. *A J Hum Genet*. 99 (1), 104-114.

Examples

```
n=1000
y = c(rep(1, n/2), rep(0, n/2))
maf = seq(0.05, 0.5, 0.05)
g = NULL
for (j in 1:10){
   geno.freq = c(maf[j]^2, 2*maf[j]*(1-maf[j]), (1-maf[j])^2)
   g = cbind(g, sample(c(0,1,2), n, replace=TRUE, prob=geno.freq))
   }
   SKATplus(y, g, X=NULL, out_type="D", tau=NULL, permutation=FALSE, B=1000)
```

SMR_interval — *Interval Estimates for Summary Data Mendelian Randomization Analysis in the Presence of Winner's Curse*

Description

`SMR_interval` calculates conservative box method interval, k-unit support interval, and Wald confidence interval for the causal effect.

Usage

```
SMR_interval(
  summary.data,
  sig.level = 5e-08,
  k = 2,
  alpha = 0.05,
  method = "box"
)
```

Arguments

`summary.data` a vector ($\hat{b}_{gx}$, se($\hat{b}_{gx}$), $\hat{b}_{gy}$, se($\hat{b}_{gy}$)) of summary data on the exposure X and the outcome Y. Due to winner's curse, the association p-value between the SNP and the exposure is less than `sig.level`.

`sig.level` the threshold p-value used to select the instrument SNP. The default is `5e-8`.

`k` the unit used for the k-unit support. The default value is 2.

`alpha` $(1 - \alpha)$ is the conservative coverage level for the box method interval or the SMR Wald interval. the default value is 0.05

`method` method to construct the interval. It is either "`support`", "`box`" or "`wald`". The default is "`box`".

Value

The returned value is method-dependent.

For `method == "box"`: A list containing the following components:

* an interval estimate.

* type of the interval: completely bounded, exclusive bounded, or bounded.

For `method == "support"`: A list containing the following components:

* Estimate The likelihood estimate of b.

* an interval estimate.

For `method == "wald"`: an interval estimate.

Author(s)

Kai Wang <kai-wang@uiowa.edu>

References

Wang, K. (2023) Support interval for two-sample summary data-based mendelia randomization. *Genes*, 14(1):211.

Wang, K. (2023) Interval estimate of causal effect in summary data based Mendelian randomization in the presence of winner's curse. *Genetic Epidemiology*, 14(1):211.

Zhu, Z. et al. (2016) Integration of summary data from GWAS and eQTL studies predicts complex trait gene targets. *Nature Genetics*, 48(5):481.

Examples

```
summary.data = c(0.13707, 0.0235162, -0.0637257, 0.013774)
SMR_interval(summary.data)
SMR_interval(summary.data, method = "support")
SMR_interval(summary.data, method = "wald")
```

tarq — *An Accurate Normalization Method for High Throughput Sequencing Data*

Description

Estimates scaling factors using the trimmed average of ratios of quantiles (TARQ) method

Usage

```
tarq(X, tau = 0.3)
```

Arguments

`X`	a matrix of raw counts. Rows are for taxa (genes, transcripts) and columns for samples
`tau`	a numerical value in (0, 0.5). The upper $\tau/2 \times 100\%$ and the lower $\tau/2 \times 100\%$ of the ratios of quantiles are trimmed

Details

Estimation of scaling factors for NGS read counts data is challenging. TARQ provides a quantile-based method for estimating scaling factors. It starts by ordering the raw counts sample by sample and constructs a reference sample from these ordered counts. To compute the scaling factor for a sample, ratios of its quantiles to those of the reference sample are formed. Zero ratios are removed. Then extreme ratios (too large or too small) are trimmed before taking average over the remaining ratios.

Value

a vector of scaling factors. Normalized counts can be obtained by `sweep(X, 2, scale.factors, FUN="/")`

Author(s)

Kai Wang <kai-wang@uiowa.edu>

References

Wang, K. (2018) An Accurate Normalization Method for Next-Generation Sequencing Data. Submitted.

Examples

```
#data(throat.otu.tab)
#data(throat.meta)
#otu.tab = t(throat.otu.tab)
#tarq(otu.tab, 0.3)

##### Use TARQ with DESeq2
#dds <- DESeqDataSetFromMatrix(countData = otu.tab,
 #                              colData = throat.meta,
 #                              design= ~ SmokingStatus)
#sizeFactors(dds) <- tarq(otu.tab, 0.3)
#dds <- DESeq(dds)
#results(dds)
#
###### Use TARQ with edgeR
#cs <- colSums(otu.tab)
#scale.factors <- tarq(otu.tab, 0.3)
#tmp <- scale.factors/cs
#norm.factors <- tmp/exp(mean(log(tmp)))
```

```
#dgList <- DGEList(counts = otu.tab, genes=rownames(otu.tab), norm.factors = norm.factors)
#designMat <- model.matrix(~ throat.meta$SmokingStatus)
#dgList <- estimateGLMCommonDisp(dgList, design=designMat)
#fit <- glmFit(dgList, designMat)
#glmLRT(fit, coef=2)
```

VSTF.test *Association Tests for Rare Variants Based on Variance-Stabilizing Transformation*

Description

VSTF.test performs tests on association between a rare variant and case-control status using a variance-stabilizing transformation.

Usage

```
VSTF.test(G, method = "Anscombe")
```

Arguments

G a 2x2 matrix. The first row is for cases and the second one for controls. In each row, the first element is the number of non-carriers and the second one is the number of carriers with at least 1 copy of the variant.

method a character string indicating which transformation to use. One of "Anscombe" (default), "arcsine", "Freeman-Tukey", and "Chanter".

Details

Each test is named after the author(s) of the corresponding publication.

Value

A list with class "test" containing the following components: * statistic the value of the test statistic. * p.value the p-value for the test computed from a chi-square distribution with 1 df. * method a character string indicting the test performed. * data.name a character string giving the name of the data.

Author(s)

Kai Wang <kai-wang@uiowa.edu>

References

Anscombe, F. J. (1948) The transformation of Poisson, binomial and negative-binomial data. *Biometrika* **35(3/4)**, 246–254.

Chanter, D. O. (1975). Modifications of the angular transformation. *Journal of the Royal Statistical Society. Series B (Applied Statistics)* **24(3)**, 354–359.

Freeman, M. F., Tukey, J. W. (1950) Transformations related to the angular and the square root. *The Annals of Mathematical Statistics* **21(4)**, 607–611.

Wang, K., Fingert, J. (2012) Statistical tests for detecting rare variants using variance-stabilizing transformations. *Annals of Human Genetics*. 76(5):402-409.

Zar, J.H. (1999) *Biostatistical Analysis, 4th ed.*, New Jersey:Prentice-Hall, Inc.

Examples

```
## Example 1 of Li et al. (2010)
G = rbind(c(14, 999), c(3, 1081))
VSTF.test(G)
VSTF.test(G, method = "arcsine")
VSTF.test(G, method = "Freeman-Tukey")
```

Index