Package 'vartools'

January 6, 2015

Type Package

Title Variant Association Tools R-package

Version 1.2.0

Date 2015-01-06

Author I.Y. Zhbannikov

Maintainer I.Y. Zhbannikov <ilyaz@uidaho.edu>

Depends survival

Description
Computes statistics of rare vatiants using CMC, KBAC, VT and other association methods

License GPL

R topics documented:

vartools-package *Variant Association tools*

Description

vartools is an R-package for the analysis of rare variants and currently has the following methods: CMC, KBAC, RVE, VT, RareCover, RBT, C-alpha.

Installation
git clone https://github.com/izhbannikov/vartools.git
R CMD INSTALL vartools

Details

Package:	vartools
Type:	Package
Version:	1.0
Date:	2014-12-10
License:	GPL (>= 2)
Project URL:	https://github.com/izhbannikov/vartools

Author(s)

Ilya Y. Zhbannikov

Maintainer: Ilya Y. Zhbannikov | i.zhbannikov@mail.ru

Examples

```
#CMC test
# Load the package
library(vartools)
?cmc

pgdata <- as.matrix(read.table(system.file("extdata","phengen.dat",package="vartools"), as.is=T, skip = 1))
cmc.pvalue <- cmc(table=pgdata)
print(cmc.pvalue)

#KBAC test
?kbac

alpha <- 0.05
num.permutation <- 3000
quiet <- 1
alternative <- 1
maf.upper.bound <- 0.05
casectrl.dat <- read.table(system.file("extdata","phengen.dat",package="vartools"), skip = 1)
kbac.pvalue <- kbac(table=casectrl.dat, alpha, num.permutation, quiet, maf.upper.bound, alternative)
print(kbac.pvalue)
```

asum *aSum statistical test*

Description

aSum, Adaptive Sum Test (Han and Pan, 2010) is a statistical test which utilizes the difference in direction of effects (protective or deleterious) of rare variants within the same genetic region analyzed by a rare variant association test.

Usage

```
asum(table, perm=100)
```

Arguments

table	a numeric matrix with first column having disease status '0' or '1' and the rest columns codes the locus genotype as '0', '1', and '2'.
perm	positive integer that defines the number of permutations. In permutation test, the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points.

Details

....

Value

A list with the following elements:

asum.stat	aSum statistic
perm.pval	permuted p-value; only when permutations is used
args	descriptive information with number of controls, cases,variants, and permutations
name	name of the statistic

References

Fang Han and Wei Pan (2010) A Data-Adaptive Sum Test for Disease Association with Multiple Common or Rare Variants. Human Heredity doi:10.1159/000288704. http://www.karger.com/doi/10.1159/000288704

Examples

```
# Load the package

library(vartools)
?asum

casectrl.dat <- read.table(system.file("extdata","phengen.dat",package="vartools"), skip = 1)
asum.stat <- asum(casectrl.dat)
asum.stat
```

calpha *C(alpha) statistical test*

Description

C(alpha) (Neale at al., 2011) is a statistical test, developed for disease traits, in order to test for the hypothesis of rare variants disease association. This test has an assumption that rare variants, which were observed in cases and controls and are a mix of various types of variants: phenotypically deleterious, protective and neutral variants.

Usage

```
calpha(table, permutations=NULL)
```

Arguments

table	a numeric matrix with first column having disease status '0' or '1' and the rest columns codes the locus genotype as '0', '1', and '2'.
permutations	positive integer that defines the number of permutations. In permutation test, the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points.

Details

....

Value

A list with the following elements:

calpha.stat	C(alpha) statistic
asym.pval	asymptotic p-value, distributed as Chi-square, with parameter degrees of freedom df=1
perm.pval	permuted p-value; only when perm is used
args	descriptive information with number of controls, cases,variants, and permutations
name	name of the statistic

References

Benjamin M. Neale, Manuel A. Rivas, Benjamin F. Voight, David Altshuler, Bernie Devlin, Marju Orho-Melander, Sekar Kathiresan, Shaun M. Purcell, Kathryn Roeder and Mark J. Daly (2011), Testing for an Unusual Distribution of Rare Variants. PLoS Genetics doi:10.1371/journal.pgen.1001322. http://dx.plos.org/10.1371/journal.pgen.1001322

Examples

```
# Load the package

library(vartools)
?calpha

casectrl.dat <- read.table(system.file("extdata","phengen.dat",package="vartools"), skip = 1)
calpha.stat <- calpha(casectrl.dat)
calpha.stat
```

---

cmc *CMC statistic implementation*

Description

This program implements the CMC statistic in [Liu and Leal 2008]. It carries out case-control association testing for rare variants for whole exome association studies. Briefly, consider a gene of length n which harbors m rare variants. Genotype on the m variant sites & the disease status (case/control) are known for each individual. The program takes as input the m-site genotype and disease status (case/control) data files, and computes a p-value indicating the significance of association. In order to speed up permutation testing we use an "adaptive" approach to obtain p-values.

Usage

```
cmc(table, method = "fisher")
```

Arguments

table	a numeric matrix with first column having disease status '0' or '1' and the rest columns codes the locus genotype as '0', '1', and '2'. DO NOT allow for missing data.
method	statistical method, Fisher test used by default

Details

....

Value

pvalue	the p-value of test.

Author(s)

Ilya Y. Zhbannikov | i.zhbannikov@mail.ru

References

Liu DJ, Leal SM, 2008 Methods for Detecting Associations with Rare Variants for Common Diseases: Application to Analysis of Sequence Data. The American Journal of Human Genetics, 83, DOI10.1016/j.ajhg.2008.06.024

Examples

```
# Load the package
library(vartools)
?cmc

pgdata <- as.matrix(read.table(system.file("extdata","phengen.dat",package="vartools"), as.is=T, skip = 1))
cmc.pvalue <- cmc(table=pgdata)
print(cmc.pvalue)
```

---

coxKM *SNP-set kernel association test for right-censored survival outcomes.*

Description

Tests for association between a set of common SNPS and a right-censored survival outcome. Warnings: (1) coxKM is meant for common genetic variants, (2) for very small p-values, it is necessary to increase no. of perturbations.

Usage

```
 coxKM(Z=NULL, U, Delta, X=NULL, gamma=NULL, kernel="linear", weights=NULL,
npert=10^4, npert.check=TRUE, npert.upper=10^8, npert.threshold=50,
impute.method = "fixed", is_check_genotype=TRUE,
is_dosage=FALSE, missing_cutoff=0.15, SetID=NULL)
```

Arguments

X	is a nxR matrix of relevant covariates with each row as a different individual and each column as a separate covariate measurement. If no additional covariates are present, X can be left unspecified or left as NULL. Note that each column of X has to be a numerical variable, non-numerical variables have to be recoded appropriately before analysis. X should not include an intercept.
Z	is a nxS numeric genotype matrix with each row as a different individual and each column as a separate snp. Each genotype should be coded as 0, 1, 2, and 9 (or NA) for AA, Aa, aa, and missing, where A is a major allele and a is a minor allele. Missing genotypes will be imputed by the simple Hardy-Weinberg equilibrium (HWE) based imputation. If kernel matrix is supplied, Z is ignored and not used in testing.
U	is a nx1 vector containing the observed times. Note: U=min(C,T) where C = censoring time, T = survival time
Delta	is a nx1 vector containing the status/event indicator.
gamma	Unless X = NULL, gamma has to be supplied. gamma is the vector of coefficients from the null cox model corresponding to X. gamma <- coxph(Surv(U,Delta)~X)$coef
kernel	Type of kernel. kernel can be an nxn kernel matrix OR one of these six options: "linear.weighted", "linear", "IBS", "IBS.weighted", "quadratic" or "2wayIX". If an nxn kernel matrix is supplied, Z is ignored and is not used in testing.

`weights` is a vector of length S of prespecified weights for the weighted kernels. Weights in coxKM are defined the same way as in SKAT. The kernel matrix of the weighted linear kernel is K=ZWWZ'.

`npert` is the number of perturbations used to calculate p-value (default =10000), npert should be at least 1000. Note that how small the p-value can be is limited by the number of perturbations. If npert.check = FALSE, the smallest possible p-value is 0.5/npert. If npert.check = TRUE, the smallest possible p-value is 0.5/(ceiling(npert.upper/10^4)*10^4). For very small p-values, to obtain accurate p-values, it is necessary to increase the number of perturbations. See npert.check.

`npert.check` TRUE/FALSE (default=TRUE). If npert.check=TRUE, coxKM first uses npert perturbations to obtain an initial p-value and checks to see if the initial p-value <= npert.threshold/npert. If the initial p-value <= npert.threshold/npert, then npert.upper perturbations is used to obtain a more accurate p-value. Setting npert.check=TRUE allows a larger number of perturbations to be used to obtain more accurate p-values only when it is necessary. For very small p-values, it may be necessary to further increase npert.upper.

`npert.upper` default=10^8. Used only if npert.check=TRUE. See npert.check.

`npert.threshold`
default=50. Used only if npert.check=TRUE. See npert.check.

`impute.method` a method to impute missing genotypes (default= "fixed"). "random" imputes missing genotypes by generating binomial(2,p) random variables (p is the MAF), and "fixed" imputes missing genotypes by assigning the mean genotype value (2p). If you use "random", you will have different p-values for different runs because imputed values are randomly assigned. Can use set.seed() to replicate results.

`is_check_genotype`
a logical value indicating whether to check the validity of the genotype matrix Z (default= TRUE). If you use non-SNP type data and want to run coxKM, please set it to FALSE. If you use SNP data or imputed data, please set it to TRUE. If is_check_genotype=FALSE, missing values in Z have to be coded only as NA since 9 will not be treated as a missing value.

`is_dosage` a logical value indicating whether the matrix Z is a dosage matrix (default= FALSE). If is_dosage=TRUE, "is_check_genotype" and "impute.method" will be ignored and coxKM will check the genotype matrix and set impute.method="fixed". Note that coxKM will also treat 9 as missing in Z.

`missing_cutoff` a cutoff of the missing rates of SNPs (default=0.15). Any SNPs with missing rates higher than cutoff will be excluded from the analysis.

`SetID` SetID.

Details

If kernel is not a matrix and Z is supplied, and either is_check_genotype=TRUE OR is_dosage=TRUE, coxKM will check the Z matrix for missing values (missing values must be coded either as NA or 9) and apply imputation. If you are using coxKM for non-SNP/dosage data, set is_check_genotype=FALSE and is_dosage=FALSE, in which case missing values must be coded as NA (9 is not considered a missing value).

Value

p.value	the p-value of coxKM based on resampling. Note that if the p-value takes on the smallest possible value based on the number of perturbations, it may be necessary to increase npert and npert.upper. See npert.check.
Q	the unscaled score test statistic of coxKM.
n.marker.test	no. of SNPs used for testing, <=S.
n.indiv	n = no. of samples
df	the estimated degrees of freedom of the test statistic (for reference only, not used in association testing) .

Author(s)

Xinyi (Cindy) Lin, Qian Zhou

References

Lin X, Cai T, Wu M, Zhou Q, Liu G, Christiani D and Lin X. 2011. Survival Kernel Machine SNP-set Analysis for Genome-wide Association Studies. Genetic Epidemiology 35:620-31. doi: 10.1002/gepi.20610

Cai T, Tonini G and Lin X. 2011. Kernel machine approach to testing the significance of multiple genetic markers for risk prediction. Biometrics, 67:975-86. doi:10.1111/j.1541-0420.2010.01544.x

Examples

```
data(examplesnpset, examplecovariates, examplephenotype1, examplephenotype2, examplephenotype3)

Z <- as.matrix(examplesnpset)
X <- as.matrix(examplecovariates)
phenotype1 <- examplephenotype1
phenotype2 <- examplephenotype2
phenotype3 <- examplephenotype3

set.seed(1)

#-----------------------------------------------------------------------
# coxKM without covariates
#-----------------------------------------------------------------------
coxKM(Z=Z, U=phenotype1$time, Delta=phenotype1$event, kernel="IBS")
coxKM(Z=Z, U=phenotype1$time, Delta=phenotype1$event, kernel="linear")
coxKM(Z=Z, U=phenotype3$time, Delta=phenotype3$event, kernel="IBS")
coxKM(Z=Z, U=phenotype3$time, Delta=phenotype3$event, kernel="linear")


#-----------------------------------------------------------------------
# coxKM with covariates
#-----------------------------------------------------------------------
Gamma <- coxph(Surv(phenotype2$time, phenotype2$event)~X)$coef
Gamma
coxKM(Z=Z, U=phenotype2$time, Delta=phenotype2$event, X=X, gamma=Gamma, kernel="IBS")
coxKM(Z=Z, U=phenotype2$time, Delta=phenotype2$event, X=X, gamma=Gamma, kernel="linear")
```

examplecovariates *Example covariates dataset for coxKM.*

Description

Example covariates dataset for coxKM.

Format

examplecovariates contains:

a numeric matrix of 2000 individuals and 2 covariates. Each row represents a different individual. coxKM.examplecovariates is identical to X in SKAT.example.

Author(s)

Xinyi (Cindy) Lin

examplephenotype1 *Example phenotype for coxKM.*

Description

Example phenotype for coxKM.

Format

examplephenotype1 contains:

a numeric matrix with two columns, the first column is the event time, the second column is the status indicator. Each row represents a different individual.

Author(s)

Xinyi (Cindy) Lin

examplephenotype2 *Example phenotype for coxKM.*

Description

Example phenotype for coxKM.

Format

examplephenotype2 contains:

a numeric matrix with two columns, the first column is the event time, the second column is the status indicator. Each row represents a different individual.

Author(s)

Xinyi (Cindy) Lin

examplephenotype3 *Example phenotype for coxKM.*

Description

Example phenotype for coxKM.

Format

examplephenotype3 contains:

a numeric matrix with two columns, the first column is the event time, the second column is the status indicator. Each row represents a different individual.

Author(s)

Xinyi (Cindy) Lin

examplesnpset *Example SNP-set for coxKM.*

Description

Example SNP-set for coxKM.

Format

examplesnpset contains:

a numeric genotype matrix of 2000 individuals and 11 SNPs. Each row represents a different individual, and each column represents a different SNP marker. coxKM.examplesnpset is subset of Z in SKAT.example.

Author(s)

Xinyi (Cindy) Lin

kbac *KBAC statistic implementation*

Description

This program implements the KBAC statistic in [Liu and Leal 2010]. It carries out case-control association testing for rare variants for whole exome association studies. Briefly, consider a gene of length n which harbors m rare variants. Genotype on the m variant sites & the disease status (case/control) are known for each individual. The program takes as input the m-site genotype and disease status (case/control) data files, and computes a p-value indicating the significance of association. In order to speed up permutation testing we use an "adaptive" approach to obtain p-values.

Usage

```
kbac(table, alpha = NULL, num.permutation = NULL, quiet = T, maf.upper.bound = 1.0, alternative = 1
```

Arguments

table	a numeric matrix with first column having disease status '0' or '1' and the rest columns codes the locus genotype as '0', '1', and '2'. DO NOT allow for missing data.
alpha	size of test, or the significant level. This feature will be useful in adaptive p-value calculation. If you do not want to use adaptive p-value, set alpha = 999 (or any number greater than 1.0).
num.permutation	number of permutations for p-value calculation. In permutation test, the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points.
quiet	when quiet = 0 the screen output would contain a summary of the KBAC test; otherwise only the p-value will be printed on screen.
maf.upper.bound	The upper bound of the MAF to be included in analysis. MAF is calculated based on observed sample. This can be arbitary although it is usually defined as 0.01 for analysis of rare variants.
alternative	Set alternative = 1 for test of deleterious variants, = 2 for test of both deleterious and protective variants. Please note that this is different from the "one/two-sided" definition in the KBAC paper.

Details

....

Value

pvalue	the p-value of test.

Author(s)

Gao Wang | wangow@gmail.com

References

Liu DJ, Leal SM, 2010 A Novel Adaptive Method for the Analysis of Next-Generation Sequencing Data to Detect Complex Trait Associations with Rare Variants Due to Gene Main Effects and Interactions. PLoS Genet 6(10): e1001156. doi:10.1371/journal.pgen.1001156

Examples

```
# Load the package
library(vartools)
?kbac

# Set parameters and use the kbac() function to obtain p-value
alpha <- 0.05
```

```
num.permutation <- 3000
quiet <- 1
alternative <- 1
maf.upper.bound <- 0.05
casectrl.dat <- read.table(system.file("extdata","phengen.dat",package="vartools"), skip = 1)
kbac.pvalue <- kbac(table=casectrl.dat, alpha, num.permutation, quiet, maf.upper.bound, alternative)
print(kbac.pvalue)

# To evaluate test at small alpha we need huge number of permutations. Adaptive approach is thus necessary.
casectrl.dat <- read.table(system.file("extdata","phengen.dat",package="vartools"), skip = 1)
kbac.pvalue <- kbac(table=casectrl.dat, 0.00001, 1000000, quiet, maf.upper.bound, alternative)
print(kbac.pvalue)

# Not using adaptive p-value calculation; will take longer time
casectrl.dat <- read.table(system.file("extdata","phengen.dat",package="vartools"), skip = 1)
kbac.pvalue <- kbac(table=casectrl.dat, 9, 1000000, quiet, maf.upper.bound, alternative)
print(kbac.pvalue)
```

`rarecover` *RareCover statistical test*

Description

Testing for rare variants with RareCover algorithm. This algorithm is similar to CMC, meaning that it follows its collapsing strategy, but uses greedy algorithm to find an optimized combination of variants in a loci for which its association signal is strongest.

Usage

```
rarecover(table, maf=0.05, dif=0.5, perm=250)
```

Arguments

`table`	a numeric matrix with first column having disease status '0' or '1' and the rest columns codes the locus genotype as '0', '1', and '2'.
`maf`	numeric value indicating the minor allele frequency threshold for rare variants
`dif`	numeric value between 0 and 1 as a threshold for the decision criterion in the RareCover algorithm
`perm`	positive integer that defines the number of permutations. In permutation test, the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points.

Details

....

Value

A list with the following elements:

rc.stat	RareCover statistic
perm.pval	permuted p-value
set	set of selected variants
args	descriptive information with number of controls, cases,variants, and permutations
name	name of the statistic

References

Bhatia G, Bansal V, Harismendy O, Schork NJ, Topol EJ, Frazer K, Bafna V (2010) A Covering Method for Detecting Genetic Associations between Rare Variants and Common Phenotypes. *PLoS Computational Biology*, **6(10)**: e1000954

Examples

```
# Load the package

library(vartools)
?rarecover

casectrl.dat <- read.table(system.file("extdata","phengen.dat",package="vartools"), skip = 1)
rarecover.stat <- rarecover(casectrl.dat)
rarecover.stat
```

---

rbt *Replication-Based statistical test*

Description

Implementation of the replication base test (RBT) proposed by Ionita-Laza et al., 2011. Replication means that it computes evidences to reject each of the two hypothesis: (1) Deleterious rare variants are enriched in cases; (2) Protective rare variants are enriched in controls.

Usage

```
rbt(table, perm=150)
```

Arguments

table	a numeric matrix with first column having disease status '0' or '1' and the rest columns codes the locus genotype as '0', '1', and '2'.
perm	positive integer that defines the number of permutations, 150 by default. In permutation test, the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points.

Details

....

Value

A list with the following elements:

rbt.stat	RBT statistic
perm.pval	permuted p-value
args	descriptive information with number of controls, cases,variants, and permutations
name	name of the statistic

References

Ionita-Laza I, Buxbaum JD, Laird NM, Lange C (2011) A New Testing Strategy to Identify Rare Variants with Either risk or Protective Effects on Disease. *PLoS Genetics*, **7(2)**: e1001289

Examples

```
# Load the package

library(vartools)
?rbt

casectrl.dat <- read.table(system.file("extdata","phengen.dat",package="vartools"), skip = 1)
rbt.stat <- rbt(casectrl.dat)
rbt.stat
```

---

skat *SKAT statistical test*

Description

SKAT (Wu at al., 2010, 2011) is a statistical test, developed for disease traits, in order to test for the hypothesis of rare variants disease association. This is a score-based variance-component test.

Usage

```
skat(table, kernel="linear", weights=NULL, a=1, b=25, permutations=NULL)
```

Arguments

table	a numeric matrix with first column having disease status '0' or '1' and the rest columns codes the locus genotype as '0', '1', and '2'.
kernel	Kernel function used. By default, a linear function is used. Other available kernels are: "wlinear", "quadratic", "IBS", "wIBS", "twowayx"
weights	a set of numeric weights for genetic variants.
a	a positive numeric value for the parameter a in the Beta distribution (a=1 by default)

b	a positive numeric value for the parameter b in the Beta distribution (b=1 by default)
permutations	positive integer that defines the number of permutations. In permutation test, the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points.

Details

....

Value

A list with the following elements:

skat.stat	SKAT statistic
asym.pval	asymptotic p-value, distributed as Chi-square, with parameter degrees of freedom df=1
perm.pval	permuted p-value; only when permutations is used
args	descriptive information with number of controls, cases,variants, and permutations
name	name of the statistic

References

Wu MC, Kraft P, Epstein MP, Taylor DM, Chanock SJ, Hunter DJ, Lin X (2010) Powerful SNP-Set Analysis for Case-Control Genome-wide Association Studies. *The American Journal of Human Genetics*, **86**: 929-942
Wu MC, 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

Examples

```
# Load the package

library(vartools)
?skat

casectrl.dat <- read.table(system.file("extdata","phengen.dat",package="vartools"), skip = 1)
skat.stat <- skat(casectrl.dat)
skat.stat
```

---

vt *Variable Threshold statistical test*

Description

This is implementation for the Variable Threshold statistical test. It assigns different weights to rare variants base on their frequencies.

Usage

```
vt(table, maf=0.05, perm=50)
```

Arguments

table	a numeric matrix with first column having disease status '0' or '1' and the rest columns codes the locus genotype as '0', '1', and '2'.
maf	numeric value indicating the minor allele frequency threshold for rare variants (must be a positive number between 0 and 1, maf=0.05 by default).
perm	positive integer that defines the number of permutations, perm=50 by default. In permutation test, the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points.

Details

....

Value

A list with the following elements:

vt.stat	VT statistic
perm.pval	permuted p-value
args	descriptive information with number of controls, cases,variants, and permutations
name	name of the statistic

References

Price AL, Kryukov GV, de Bakker PIW, Purcell SM, Staples J, Wei LJ, Sunyaev SR (2010) Pooled Association Tests for Rare Variants in Exon-Sequencing Studies. *The American Journal of Human Genetics*, **86**: 832-838

Examples

```
# Load the package
library(vartools)
?vt

casectrl.dat <- read.table(system.file("extdata","phengen2.dat",package="vartools"), skip = 1)
vt.stat <- vt(casectrl.dat)
vt.stat
```

wss *WSS statistical test*

Description

WSS (Madsen and Browning, 2009) is a statistical test, developed for disease traits, in order to test for the hypothesis of rare variants disease association. WSS test introduces the method of assigning "weights" to rare variants found in a genetic region before they are collapsed.

Usage

```
wss(table, perm=100)
```

Arguments

table	a numeric matrix with first column having disease status '0' or '1' and the rest columns codes the locus genotype as '0', '1', and '2'.
perm	positive integer that defines the number of permutations. In permutation test, the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points.

Details

....

Value

A list with the following elements:

wss.stat	WSS statistic
perm.pval	permuted p-value; only when permutations is used
args	descriptive information with number of controls, cases,variants, and permutations
name	name of the statistic

References

Bo Eskerod Madsen and Sharon R. Browning (2009) A Groupwise Association Test for Rare Mutations Using a Weighted Sum Statistic. PLoS Genetics doi:10.1371/journal.pgen.1000384. http://dx.plos.org/10.1371/journal.pgen.1000384

Examples

```
# Load the package

library(vartools)
?wss

casectrl.dat <- read.table(system.file("extdata","phengen.dat",package="vartools"), skip = 1)
wss.stat <- wss(casectrl.dat)
wss.stat
```

Index