Package 'LSKAT'

January 13, 2017

Type Package

Title SNP-set (Sequence) Kernel Association Test for longitudinal data.

Version 0.5

Date 2016-10-20

Author Zhong Wang and Zuoheng Wang

Maintainer Zhong Wang <zhong.wang@yale.edu>

Description Kernel based SNP set test for longitudinal data.

License GPL (>= 2)

Depends R (>= 2.12.1), SKAT, mvtnorm, snpStats, snowfall

Suggested sn

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-10-20

R topics documented:

`longskat_est_model` *NULL model estimation*

Description

Estimating the parameters and residuals for the NULL model in LSKAT.

Usage

```
longskat_est_model( phe.long,
    phe.cov,
    phe.time  = NULL,
    time.cov  = 0,
    intercept = FALSE,
    method    = c("REML", "ML"),
    g.maxiter = 20,
    par.init  = list(),
    verbose   = F)
```

Arguments

`phe.long`	Phenotype matrix with *m* rows denoting the individuals and *n* columns denoting the time points, the row name indicates the individual's ID.
`phe.cov`	Time covariate matrix with *m* rows denoting individuals and *x* columns denoting the covariate variables, the row name indicates the individual's ID.
`phe.time`	Time point matrix with *m* rows denoting individuals and *n* columns denoting the time points, the row name indicates the individual's ID. If this matrix is not specified, the default matrix is generated.
`time.cov`	Numeric, indicating whether the time exponnents are included as extra covariates, The time points are used if 1, the time points and time squares are used if 2, and so on. The default value (0) doesn't use the time covariate.
`intercept`	Logical variable, indicating whether the intercept is estimated.
`method`	String, REML or ML are available for the parameter estimation.
`g.maxiter`	Numeric, the maximum count for the iterative estimation.
`par.init`	List, the initial values for the parameter `rho`, `sig.a`, `sig.b`, `sig.e`.
`verbose`	Logical variable, indicating whether some debug information can be outputted.

Value

This function returns an list object with model parameters and residuals of the NULL model which assumes there is no association between genes and longitudinal phenotypes.

The return object is a list with the following items:

`par`	List, model paramters as shown in below.
`likelihood`	Numeric, the likelihood value estimated by REML or ML.

`phe.delt`	Residual matrix with the row name indicating the individual's ID, the structure is same as `phe.long`.
`phe.time`	Time matrix, copied from the input parameter `phe.time`.
`phe.cov`	Covariate matrix, copied from the input parameter `phe.cov`.

The Model paramters: `par` has the following sub-items:

`intercept`	Logical variable copied from the input parameter, indicating the intercept is estimated.
`mu`	Numeric indicating the intercept value.
`cov.effect`	String indicating the coefficient of the covariates except intercept
`sig.a`	String indicating the standar deviation of individual random effects
`sig.b`	String indicating the standar deviation of individual-specific timede-pendent random effects.
`sig.e`	String indicating the standar deviation of measurement error.
`rho`	String indicating the corelation coefficient of covariance structure.
`time.cov`	Numeric, indicating whether consider times as covariate, 0 means no time effects, 1 means time effects, 2 means time effects and time square effects are included as covariates. and so on.
`time.effect`	Vector of numeric, the time coefficient of time effects. The 1st item is the coefficient for time effects, The 2nd item is the coefficient for time square effects and so on.

After obtaining the model parameters, please use the `longskat_gene_test` to test the association between gene and traits.

References

Wang Z., Xu K., Zhang X., Wu X., and Wang Z., (2016) Longitudinal SNP-set association analysis of quantitative phenotypes. Genetic Epidemiology.

Examples

```
## Data simulation using the default parameters
p0 <- longskat_gene_simulate();

## Estimating the model parameters and residuals
r.model0 <- longskat_est_model( p0$phe.long, p0$phe.cov, g.maxiter=3, verbose=T);

##print this model
print(r.model0);
```

`longskat_gene_plink` *LSKAT test for plink data set.*

Description

This function provides a pipeline for the plink data set based on the LSKAT function. Except the plink data set(BED, BIM and FAM), SNP set table is same as SKAT package.

Usage

```
longskat_gene_plink(file.plink.bed,
    file.plink.bim,
    file.plink.fam,
    file.phe.long,
    file.phe.cov,
    file.phe.time = NULL,
    file.gene.set,
    gene.set = NULL,
    options = list())
```

Arguments

`file.plink.bed`	File name, PLINK bed file
`file.plink.bim`	File name, PLINK bim file
`file.plink.fam`	File name, PLINK fam file
`file.phe.long`	File name, indicating phenotype file in CSV format with row name (individual ID) and header information (Measured Index). Each individual is encoded to one row data which has individual ID as row name and multiple observed values followed by row name.
`file.phe.cov`	File name, indicating covariate file in CSV format with row name (individual ID) and header information (Covariate Index). Each individual is encoded to one row data which has individual ID as row name and covariate values followed by row name.
`file.phe.time`	File name, indicating measured times in CSV format with row name (individual ID) and header information (Measured Time). Each individual is encoded to one row data which has individual ID as row name and covariate values followed by row name.
`file.gene.set`	File name, indicating gene set table which has two columns, 1st column is gene name, 2nd column is variant name and no header.
`gene.set`	numeric or string. indicating the gene name or gene index in the gene set table.
`options`	String, see the details.

Details

The gene set file is same as the SetID file in SKAT packa which is white-space (space or tab) seperated file with 2 columns: SetID (1st column) and SNP_ID (2nd column), Please keep in mind that there should be no header!

The phenotype file, covariate file and measured time file are CSV files with header and row name(indicating the individual id), these 3 files should have same individual order. Please note that the columns in measured time file are corresponding to same columns in the phenotype file.

The `options` list has some important feature to tune the association test. The default values are defined in the package as the follows:

```
options <- list( rare.cutoff = NULL,
                 time.cov      = 0,
                 g.maxiter     = 20,
                 weights.common= c(0.5,0.5),
                 weights.rare  = c(1,25),
                 run.cpp       = F,
                 verbose       = F,
                 n.cpu         = 1,
                 snp.impute    = "mean",
                 intercept     = F,
                 plink.path    = NULL,
                 test.type     = "Joint",
                 est.method    = "REML");
```

rare.cutoff Numeric, a value of MAF cutoff for the rare SNPs. Only SNPs that have MAFs smaller than this are considered as rare SNP. The default criterion of rare SNP is calculated by the formula $1/\sqrt{2 * sample}$

time.cov Numeric, indicating which order of time is included as extra covariates. If this value is 2, time and time square are considered as extra covariates.

g.maxiter Number, indicating the number of maximum iteration is applied to MLE alogrithm.

weights.common a numeric vector of parameters of beta weights for common variants (default=c(0.5,0.5)).

weights.rare a numeric vector of parameters of beta weights for rare variants (default=c(1,25)).

run.cpp Logical, indicating whether C/C++ functions are used to compute LSKAT.

verbose Logical, indicating the computaional process output much more information than normal mode.

snp.impute String, indicating the method of SNP imputation, the default model uses the mean of each variant to replace the missing SNP data. Two optional values: 'mean' or 'random'.

intercept Logical, indicating whether the intercept is considered in the NULL model.

plink.path String, indicating PLINK command path, for the large PLINK data, the package will load small plink data set extracted by the plink command rather than the whole data set.

test.type String, Three models can be selected, "joint", "Common.Only", "rare.Only".

est.method String, indicating the estimate method for NULL model, two options, REML or ML, default is REML.

Value

A list object with 3 items is returned by this function:

`par`	The parameters of this function call.
`mle`	The estimation of NULL Model returned by `longskat_est_model`, including model parameters and residuals, please refer to the structure in `longskat_est_model`.
`result`	Matrix containing the test result of LSKAT and L-burden for all genes, the sub-items are described as follows.

The structure: of `result` item:

`index`	Number, gene index
`gene.name`	String, gene name
`chr`	Number, chromosome
`min.pos`	Number, minimum position of all variants
`snp.total`	Number, the total number of variants
`snp.rare`	Number, the number of rare variants.
`q.lskat`	Numeric, the statistical test of LSKAT
`p.lskat`	Numeric, the p-value of LSKAT
`q.burden`	Numeric, the statistical test of L-Burden test
`p.burden`	Numeric, the p-value of L-Burden test

References

Wang Z., Xu K., Zhang X., Wu X., and Wang Z., (2016) Longitudinal SNP-set association analysis of quantitative phenotypes. Genetic Epidemiology.

See Also

`longskat_gene_test` and `longskat_est_model`

Examples

```
## data simulation for the test of type 1 error
p0 <- longskat_gene_simulate( plink.format=T, file.plink.prefix="tmp-gene-plink",
      power.test=F, n.gene=20);

## test all genes in the PLINK data set
r.lskat0 <- longskat_gene_plink(p0$file.plink.bed, p0$file.plink.bim, p0$file.plink.fam,
      p0$file.phe.long, p0$file.phe.cov, NULL,
      p0$file.gene.set, gene.set=c(5:10), options=list(verbose=T) );

## print the results
r.lskat0;

## data simulation for the power test
p0 <- longskat_gene_simulate( plink.format=T, file.plink.prefix="tmp-gene-plink",
      power.test=T, n.gene=20);
```

```
## test all genes in the PLINK data set
r.lskat1<-longskat_snp_plink( p0$file.plink.bed, p0$file.plink.bim, p0$file.plink.fam,
      p0$file.phe.long, p0$file.phe.cov, file.gene.set=p0$file.gene.set,
      snp.set=c(1:199),
      options=list(time.cov=1, g.maxiter=6, n.cpu=2, test.type="Rare.only" ));

## print the results
r.lskat1;

#draw the Manhattan figure for the result
plot( r.lskat1, pdf.file="tmp-plink-lskat1.pdf", title="TEST PLINK 1", bonferroni=T);
```

`longskat_gene_simulate`
Simulation for LSKAT test

Description

Using the pre-defined parameters to make the simulation data for the LSKAT test (including power test and type I error test).

Usage

```
longskat_gene_simulate( power.test = TRUE,
    n.minsect = 3000,
    n.maxsect = 30000,
    n.sample = 800,
    n.time = 6,
    n.gene = 10,
    plink.format = FALSE,
    file.plink.prefix = "LSKAT.plink.test",
    geno.miss = 0.01,
    pheno.miss = 0.1,
    pheno.dist = "mn",
    pheno.cov = "AR1",
    intercept = FALSE,
    par = list() )
```

Arguments

`power.test`	Logical variable, indicating whether simulate individual random effects and the individual-specific timede-pendent random effects for the pwer test, otherwise, FALSE indicates type I error test.
`n.minsect`	Numeric, the minimum size of gene(Unit: BP)
`n.maxsect`	Numeric, the maximum size of gene(Unit: BP)
`n.sample`	Numeric, sample size, ie, individual count.
`n.time`	Numeric, measurement time.
`n.gene`	Numeric, gene number. If simulation for power test, the 1st gene is the causal gene, the rest are non-causal gene.

`plink.format` Logical variable, indicating whether the data will be stored into PLINK file in addtionalto return a list obecjt with multiple matrices.

`file.plink.prefix` String, the prefix file name for plink data set if plink.format is TRUE.

`geno.miss` Numeric, the missing rate for genome data set.

`pheno.miss` Numeric, the missing rate for phenotype traits.

`pheno.dist` String, the distribution of individual-specific timede-pendent random effects, four optional values: 'mn', 'mt', 'msn', 'mmn', see *details*.

`pheno.cov` String, the covariance structure of individual-specific timede-pendent random effects, three optional values: 'AR1', "SAD1' and 'CS', see *details*.

`intercept` Logical variable, indicating whether intercept is used in phenotypic traits.

`par` List, the parameters for the phenotype traits, including covariates and individual-specific timede-pendent random effects.

Details

The simaltion is generated by the following formula:

$$Y_{ij} = intercept + b1 * X1_{ij} + b2 * X2_{ij} + a_i + r_{ij} + e_{ij}$$

a_i:individual random effects

r_{ij}:individual-specific timede-pendent random effects

e_{ij}:measurement error

the individual random effects follow the normal distribution with the standard deviation `sig.a`.

the individual-specific timede-pendent random effects follow the multivariate normal distribution with covariance structure: AR1, SAD1 or CS.

the individual random effects follows the distribution of t, normal, skew normal or mixed normal.

The covariance structure:

`AR1` first-order Autoregressive model [AR(1)], parameters: `par$rho` and `par$sig.b`

`SAD1` first-order structured antedependence [SAD(1)], parameters: `par$rho` and `par$sig.b`

`CS` compound symmetry model, parameters: `par$rho` and `par$sig.b`

The distibution of measurement error:

`mn` Normal distribution, parameters: `par$sig.e`

`mt` Student distribution, parameters: df=10

`msn` Skew normal distribution, parameters: `par$sig.e`, alpha = 40

`mmn` Mixed normal distribution,parameters: `par$par.e[1]`, `par$par.e[2]`, `par$par.e[3]`

The pre-defined parameters in the package have the following values:

```
par <- list(b0=1, b1=0.5, b2=0.5,
    sig.a=0.8, sig.b=0.8, sig.e=0.8,
    rho=0.7,
    cov.param   = c(0,1, 0.1),
    time.cov    = 0,
    time.effect = c(0.2, -0.08),
    max.common.causal  = 4,
    coef.common.causal = 0.12,
    max.rare.causal    = 10,
    coef.rare.causal   = 0.08,
    positive.ratio     = 1,
    rare.cutoff        = 0.05 );
```

`b0` Numeric, the intercept value if the intercept is enable.

`b1` Numeric, the coefficient of the 1st covariate, binary variable.

`b2` Numeric, the coefficient of the 2nd covariate, continuous variable.

`sig.a` Numeric, the standar deviation of individual random effects.

`sig.b` Numeric, the standar deviation of individual-specific timede-pendent random effects.

`sig.e` Numeric, the standar deviation of measurement error.

`rho` Numeric, the corelation coefficient of covariance structure.

`cov.param` Vector, the other parameters of covariance structure except `rho`.

`time.cov` Numeric, indicating whether consider times as covariate, 0 means no time effects, 1 means time effects, 2 means time effects and time square effects are included as covariates. and so on.

`time.effect` Numeric, the time coefficient of time effects. The 1st item is the coefficient for time effects, The 2nd item is the coefficient for time square effects and so on.

`max.common.causal` Numeric, the maximum number of common causal SNPs.

`coef.common.causal` Numeric, the effect coefficient for common causal SNPs.

`max.rare.causal` Numeric, the maximum number of rare causal SNPs.

`coef.rare.causal` Numeric, the effect coefficient for rare causal SNPs.

`positive.ratio` Numeric, the positive ratio in all causal SNPs.

`rare.cutoff` Numeric, hard cuf off for rare MAF, default rare cut off is calculated by the formula: $1/\sqrt{2 * sample}$.

Value

A list object is returned with the following items:

`file.plink.bed`
: String, if `plink.format` is assigned to TRUE, this is the name of the PLINK file containing the packed binary SNP genotype data. It should have the extension .bed.

`file.plink.bim`
: String, if `plink.format` is assigned to TRUE, this is the name of the PLINK file containing the SNP descriptions.

`file.plink.fam`
: String, if `plink.format` is assigned to TRUE, this is the name of the PLINK file containing subject (and, possibly, family) identifiers.

`file.gene.set`
: String, if `plink.format` is assigned to TRUE, this is the name of the table file contaning the gene defintion, 1st column is gene and 2nd column is SNP name.

`file.phe.cov`
: String, if `plink.format` is assigned to TRUE, this is the CSV file containing covariate matrix with m rows (individuals) and n columns (covariates), and also with the individual IDs as row names.

`file.phe.long`
: String, if `plink.format` is assigned to TRUE, this is the CSV file containing phenotype traits matrix with m rows (individuals) and n columns (covariates), and also with the individual IDs as row names.

`phe.long`
: Matrix, phenotype traits matrix with m rows (individuals) and n columns (covariates), and also with the individual IDs as row names.

`phe.cov`
: Matrix, covariate matrix with m rows (individuals) and n columns (covariates), and also with the individual IDs as row names.

`snp.mat`
: List, containing multiple matrices, each matrix includes all SNPs in the gene.

References

Wang Z., Xu K., Zhang X., Wu X., and Wang Z., (2016) Longitudinal SNP-set association analysis of quantitative phenotypes. Genetic Epidemiology.

Examples

```
## data simulation for the power test
p0 <- longskat_gene_simulate( plink.format=T, file.plink.prefix="tmp-plink-simulate",
      power.test=T );

## test all genes in the PLINK data set
r.lskat1 <- longskat_gene_plink(p0$file.plink.bed, p0$file.plink.bim, p0$file.plink.fam,
      p0$file.phe.long, p0$file.phe.cov, NULL, p0$file.gene.set, options=list(g.maxiter=3


## data simulation for the test of type 1 error
p1 <- longskat_gene_simulate( plink.format=T, file.plink.prefix="tmp-plink-simulate",
      power.test=F );

## test all genes in the PLINK data set
r.lskat2 <- longskat_gene_plink(p1$file.plink.bed, p1$file.plink.bim, p1$file.plink.fam,
      p1$file.phe.long, p1$file.phe.cov, NULL, p1$file.gene.set,
      options=list(g.maxiter=3, plink.path="plink"));
```

`longskat_gene_test` *Assoictaion test using LSKAT*

Description

Assoictaion test for the comined effect of common and rare variants using LSKAT

Usage

```
longskat_gene_test( r.model,
    snp.mat,
    weights.common=c(0.5,0.5),
    weights.rare=c(1,25),
    rare.cutoff=NULL,
    test.type="Joint",
    snp.impute = "mean",
    run.cpp=F,
    verbose=F)
```

Arguments

`r.model`	The list object obtained from the function `longskat_est_model`, including the estimated parameters and residuals.
`snp.mat`	Matrix with m row for individuals and n columns for the variaints(SNPs), also with the individuals' ID as the row names.
`weights.common`	a numeric vector of parameters of beta weights for common variants (default=c(0.5,0.5)).
`weights.rare`	a numeric vector of parameters of beta weights for rare variants (default=c(1,25)).
`rare.cutoff`	Numeric, a value of MAF cutoff for the rare SNPs. Only SNPs that have MAFs smaller than this are considered as rare SNP. The default criterion of rare SNP is calculated by the formula $1/\sqrt{2 * sample}$
`test.type`	String, Three models can be selected, "joint", "Common.Only", "rare.Only".
`snp.impute`	String, indicating the method of SNP imputation, the default model uses the mean of each variant to replace the missing SNP data.
`run.cpp`	Logical, indicating whether C/C++ functions are used to compute LSKAT.
`verbose`	Logical variable, indicating whether some debug information can be outputted.

Value

The list object is returned by this function with the following items:

`snp.NMISS`	Vector, the missing rate for each SNP.
`snp.MAF`	Vector, the MAF for each SNP.
`snp.total`	Numeric, the total number of variants.
`snp.rare`	Numeric, the number of rare variants.
`q.lskat`	Numeric, the statistical test of LSKAT

p.lskat	Numeric, the p-value of LSKAT
q.burden	Numeric, the statistical test of L-Burden Test
p.burden	Numeric, the p-value of L-Burden Test

References

Wang Z., Xu K., Zhang X., Wu X., and Wang Z., (2016) Longitudinal SNP-set association analysis of quantitative phenotypes. Genetic Epidemiology.

See Also

longskat_est_model

Examples

```
## data simulation for the power test
p0 <- longskat_gene_simulate( plink.format=T, file.plink.prefix="tmp-gene-test",
      power.test=T, n.gene=5 );

## model estimation
r.reml <- longskat_est_model( p0$phe.long, p0$phe.cov, phe.time = NULL, g.maxiter=3,
      method="REML", verbose=T )

print(r.reml);

## test the genes using the function 'longskat_gene_test'
for(i in 1:length(p0$snp.mat))
{
    r.lskat0 <- longskat_gene_test(r.reml, p0$snp.mat[[i]], snp.impute="mean");
    print(r.lskat0);
}
```

---

longskat_get_gene *Get SNP matrix of gene.*

Description

This function provides a interface to read all SNPs of genes from the PLINK data object.

Usage

```
longskat_get_gene(  gen.obj, gene.set, snp.impute="mean", verbose = FALSE )
```

Arguments

gen.obj	Reference class , it is a wrapper of plink data object obtained from the function longskat_plink_load.
gene.set	Vector of numeric or string, indicating the gene index or gene name.
snp.impute	String, indicating the method of SNP imputation, the default model uses the mean of each variant to replace the missing SNP data. Two optional values: 'mean' or 'random'.

verbose	Logical variable, indicating whether some debug information can be outputted. ;

Details

The parameter gene.set can be gene index (numeric) or gene names (string).

Value

The returned list object could contain multiple genes. It includes

snp.mat	List, SNP Matrices for genes in gene.set.
maf	List, SNP MAF vectors for genes in gene.set.
nmiss	List, SNP missing information for genes in gene.set.
gene.name	List, gene names for genes in gene.set.

References

Wang Z., Xu K., Zhang X., Wu X., and Wang Z., (2016) Longitudinal SNP-set association analysis of quantitative phenotypes. Genetic Epidemiology.

See Also

longskat_plink_load and longskat_gene_test

Examples

```
## data simulation for the power test
p0 <- longskat_gene_simulate( plink.format=T, file.plink.prefix="tmp-plink-load",
      power.test=T, n.gene=20);

## Loading PLINK data set into the reference object
plk <- longskat_plink_load ( p0$file.plink.bed, p0$file.plink.bim, p0$file.plink.fam,
      p0$file.gene.set,  verbose=TRUE);

## Retriving the SNP matrix from the reference object
snps <- longskat_get_gene( plk, 1:2, snp.impute="mean" );

## show the structures of the result.
str(snps);
```

---

longskat_plink_load

Loading plink data set.

Description

This function provides a pipeline for the plink data set based on the LSKAT function. Except the plink data set(BED, BIM and FAM), SNP set table is same as SKAT package.

Usage

```
longskat_plink_load(file.plink.bed,
    file.plink.bim,
    file.plink.fam,
    file.gene.set,
    plink.path=NULL,
    verbose=FALSE)
```

Arguments

`file.plink.bed`	File name, PLINK bed file, containing the packed binary SNP genotype data
`file.plink.bim`	File name, PLINK bim file, containing the SNP descriptions
`file.plink.fam`	File name, PLINK fam file, containing subject(and, possibly, family) identifiers
`file.gene.set`	File name, indicating SNP set table which has two columns, gene name in 1st column and variant name in 2nd column and no header.
`plink.path`	String, indicating PLINK command path, for the large PLINK data, the package will load partial plink data extracted by the plink command rather than the whole data set.
`verbose`	Logical Variable, indicating whether some information are outputted for debug.

Details

The SNP set file is same as the `SetID` file in *SKAT* package which is white-space (space or tab) seperated file with 2 columns: SetID (1st column) and SNP_ID (2nd column), Please keep in mind that there should be *no header!*

Value

This function returns a reference class (`"'PLINK.refer'"`)for PLINK data operation which provides a interface to access the SNP information. `longskat_get_gene` use this object to extract the SNP matrix for genes.

References

Wang Z., Xu K., Zhang X., Wu X., and Wang Z., (2016) Longitudinal SNP-set association analysis of quantitative phenotypes. Genetic Epidemiology.

See Also

`longskat_get_gene`

Examples

```
## see the example in the function longskat_get_gene
```

`longskat_snp_plink` *Single SNP associaion test using LSKAT for PLINK data set.*

Description

This function provides a pipeline for the plink data set based on the LSKAT function. Except the plink data set(BED, BIM and FAM), SNP set table is same as SKAT package.

Usage

```
longskat_snp_plink(file.plink.bed,
    file.plink.bim,
    file.plink.fam,
    file.phe.long,
    file.phe.cov,
    file.phe.time = NULL,
    file.gene.set=NULL,
    snp.set = NULL,
    options = list(),
    verbose=FALSE);
```

Arguments

`file.plink.bed`	File name, PLINK bed file
`file.plink.bim`	File name, PLINK bim file
`file.plink.fam`	File name, PLINK fam file
`file.phe.long`	File name, indicating phenotype file in CSV format with row name (individual ID) and header information (Measured Index). Each individual is encoded to one row data which has individual ID as row name and multiple observed values followed by row name.
`file.phe.cov`	File name, indicating covariate file in CSV format with row name (individual ID) and header information (Covariate Index). Each individual is encoded to one row data which has individual ID as row name and covariate values followed by row name.
`file.phe.time`	File name, indicating measured times in CSV format with row name (individual ID) and header information (Measured Time). Each individual is encoded to one row data which has individual ID as row name and covariate values followed by row name.
`file.gene.set`	File name, indicating gene set table which has two columns, 1st column is gene name, 2nd column is variant name and no header.
`snp.set`	numeric or string. indicating the gene name or gene index in the gene set table
`options`	String, see the *details*
`verbose`	Logical variable, indicating whether some debug information can be outputted.

Details

None

Value

A list object with 3 items is returned by this function:

`par`	The parameters of this function call.
`mle`	The estimation of NULL Model returned by `longskat_est_model`, including model parameters and residuals, please refer to the structure in `longskat_est_model`.
`result`	Matrix containing the test result of LSKAT for all SNPs, the sub-items are described as follows.

The structure: of `result` item:

`index`	Number, gene index
`snp.name`	String, snp name
`gene.name`	String, gene name
`chr`	Number, chromosome
`pos`	Number, position
`MAF`	Number, Minor Allel Frequency.
`NMISS`	Number, number of missing SNP.
`rare`	Logical, rare SNP or not.
`q.lskat`	Numeric, the statistical test of LSKAT
`p.lskat`	Numeric, the p-value of LSKAT

Examples

```
## data simulation for the power test.
p1 <- longskat_gene_simulate( plink.format=T, file.plink.prefix="tmp-simu-snp",
      power.test=T, n.gene=2);

## Single SNP test by LSKAT
r.lskat <- longskat_snp_plink( p1$file.plink.bed, p1$file.plink.bim, p1$file.plink.fam,
      p1$file.phe.long, p1$file.phe.cov, NULL, p1$file.gene.set );

## show the result
print(r.lskat);

## draw the result in the Manhattan figure.
plot(r.lskat, "tmp-simu-snp.pdf", "title")
```

`longskat_snp_test` *None*

Description

None

Usage

```
longskat_snp_test(r.model,
    snp,
    weights.common = c(0.5, 0.5),
    weights.rare = c(1, 25),
    snp.impute = "mean",
    rare.cutoff = NULL,
    run.cpp = F,
    verbose = FALSE )
```

Arguments

`r.model`	The list object obtained from the function `longskat_est_model`, including the estimated parameters and residuals.
`snp`	Matrix with m row for individuals and n columns for the variaints(SNPs), also with the individuals' ID as the row names.
`weights.common`	a numeric vector of parameters of beta weights for common variants (default=c(0.5,0.5)).
`weights.rare`	a numeric vector of parameters of beta weights for rare variants (default=c(1,25)).
`rare.cutoff`	Numeric, a value of MAF cutoff for the rare SNPs. Only SNPs that have MAFs smaller than this are considered as rare SNP. The default criterion of rare SNP is calculated by the formula $1/\sqrt{2 * sample}$
`run.cpp`	Logical, indicating whether C/C++ functions are used to compute LSKAT.
`verbose`	Logical variable, indicating whether some debug information can be outputted.

Value

None

Examples

```
## data simulation for the power test
p0 <- longskat_gene_simulate( plink.format=F, power.test=T, n.gene=1);

## model estimation
r.model <- longskat_est_model( p0$phe.long, p0$phe.cov, g.maxiter=3, verbose=T);

for(i in 1:NCOL(p0$snp.mat[[1]]))
{
    ## test all SNPs in the 1st gene.
    r.lskat <- longskat_snp_test(r.model, p0$snp.mat[[1]][,i]);
    print(r.lskat);
```

```
}
```

`plot.LSKAT.gen.plink` *Plot Manhattan figure.*

Description

This function draws manhattan figure using the p-value of LSKAT and L-Burden.

Usage

```
plot.LSKAT.gen.plink( r.lskat, pdf.file, title, y.max, bonferroni )
```

Arguments

`r.lskat`	The list object with S3 class name: LSKAT.gen.plink, obtained from `longskat_gene_plink`.
`pdf.file`	String, the file name of PDF output.
`title`	String, title
`y.max`	Numeric, the maximum value of Y-axis
`bonferroni`	Boolean, indicating whether Bonferroni correction is used in the plot

See Also

`longskat_gene_plink`

Examples

```
## check the code in the function longskat_gene_plink
```

`plot.LSKAT.snp.plink` *Plot Manhattan figure using SNP association test*

Description

This function draws manhattan figure using the p-value of LSKAT and L-Burden.

Usage

```
plot.LSKAT.snp.plink(r.lskat, pdf.file, title, y.max)
```

Arguments

r.lskat	The list object with S3 class name: LSKAT.gen.plink, obtained from longskat_snp_plink.
pdf.file	String, the file name of PDF output.
title	String, title
y.max	Numeric, the maximum value of Y-axis
bonferroni	Boolean, indicating whether Bonferroni correction is used in the plot

See Also

longskat_snp_plink

Examples

```
## check the code in the function longskat_snp_plink
```

---

summary.LSKAT.gen.plink

Summarizing the LSKAT results for all genes

Description

Summarizing the LSKAT results for all genes in PLINK data set

Usage

```
summary.LSKAT.gen.plink(r.lskat)
```

Arguments

r.lskat	The list object with S3 class name: LSKAT.gen.plink, obtained from longskat_gene_plink.

See Also

longskat_gene_plink

Examples

```
## data simulation for the power test
p0 <- longskat_gene_simulate( plink.format=T, file.plink.prefix="tmp-gene-plink",
      power.test=T );

## LSKAT test
r.lskat1 <- longskat_gene_plink(p0$file.plink.bed, p0$file.plink.bim, p0$file.plink.fam,
      p0$file.phe.long, p0$file.phe.cov, NULL, p0$file.gene.set,
      options=list(g.maxiter=3, plink.path="plink", est.method = "ML"), verbose=T );

## sumamry information, this function
df <- summary(r.lskat1);

## check the sumamry information
show(df);
```

summary.LSKAT.snp.plink
Summarizing the LSKAT results for all SNPs

Description

Summarizing the LSKAT results for all SNPs in PLINK data set

Usage

```
summary.LSKAT.snp.plink(r.lskat)
```

Arguments

r.lskat	The list object with S3 class name: LSKAT.snp.plink, obtained from longskat_snp_plink.

See Also

longskat_snp_plink and summary.LSKAT.gen.plink

Index