Package 'SKAT'

May 7, 2026

Type Package

Title SNP-Set (Sequence) Kernel Association Test

Version 2.2.5

Date 2023-01-12

Author Seunggeun (Shawn) Lee and Zhangchen Zhao, with contributions from Larisa Miropolsky and Michael Wu

Maintainer Seunggeun (Shawn) Lee <lee7801@snu.ac.kr>

Description Functions for kernel-regression-based association tests including Burden test, SKAT and SKAT-O. These methods aggregate individual SNP score statistics in a SNP set and efficiently compute SNP-set level p-values.

License GPL (>= 2)

Depends R (>= 2.13.0), Matrix, SPAtest, RSpectra

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-01-15 15:50:09 UTC

Contents

`Close_SSD` *Close SNP set data file (SSD)*

Description

Close the opened SNP Set data file (SSD). After using the SSD file, it must be closed.

Usage

```
Close_SSD()
```

Author(s)

Seunggeun Lee, Larisa Miropolsky

`Generate_SSD_SetID` *Generate SNP set data file (SSD)*

Description

Generate a SNP set data file (SSD) from binary plink data files using user specified SNP sets.

Usage

```
Generate_SSD_SetID(File.Bed, File.Bim, File.Fam, File.SetID,
 File.SSD, File.Info, Is.FlipGenotype=TRUE)
```

Arguments

File.Bed	name of the binary ped file (BED).
File.Bim	name of the binary map file (BIM).
File.Fam	name of the FAM file (FAM).
File.SetID	name of the SNP set ID file that defines SNP sets. The first column must be Set ID, and the second column must be SNP ID. There should be no header!!
File.SSD	name of the SSD file generated.
File.Info	name of the SSD info file generated.
Is.FlipGenotype	internal use only, please do not change

Details

The SetID file is a white-space (space or tab) delimitered file with 2 columns: SetID and SNP_ID.

Please keep in mind that there should be no header! The SNP_IDs and SetIDs should be less than 50 characters, otherwise, it will return an error message.

The SSD file is a binary formated file with genotypes. The SSD info file is a text file with general information on data and SNP sets (first 6 rows), and information on each set (after 8th row).

Author(s)

Seunggeun Lee, Larisa Miropolsky

Get_EffectiveNumberTest

Estimate the effective number of tests for Bonferroni correction

---

Description

Estimate the effective number of tests for Bonferroni correction using the minimum achievable p-values (MAP).

Usage

```
Get_EffectiveNumberTest(MAP, alpha=0.05, Is.MidP=TRUE)
```

Arguments

MAP	a vector of the minimum achievable p-values (MAP).
alpha	a significant level.
Is.MidP	a logical value indicating whether p-values are mid-p-values.

Value

Effective number of test for Bonferroni correction at the level alpha. MAP can be obtained from SKATBinary functions.

Author(s)

Seunggeun Lee

References

Lee, S., Fuchsberger, C., Kim, S., Scott, L. (2015) An efficient resampling method for calibrating single and gene-based rare variant association analysis in case-control studies. *Biostatistics*, in press.

`Get_Genotypes_SSD` *Get Genotype data from SSD file*

Description

Read a SSD file and return a genotype matrix.

Usage

```
Get_Genotypes_SSD(SSD_INFO, Set_Index, is_ID = TRUE)

Get_Genotypes_SSD_Sparse(SSD_INFO, Set_Index)
```

Arguments

`SSD_INFO`	SSD_INFO object returned from Open_SSD.
`Set_Index`	a numeric value of Set index. The set index of each set can be found from SetInfo object of SSD.INFO.
`is_ID`	a logical value indicating whether to read SNP ID (default=TRUE). If TRUE, it reads SNP IDs and use them as column names.

Value

A genotype matrix with n rows and m columns, where n is the number of samples and m is the number of SNPs. Get_Genotypes_SSD_Sparse returns a sparse matrix. Get_Genotypes_SSD_Sparse always returns SNP IDs as the column names, so does not have is_ID parameter.

Author(s)

Seunggeun Lee, Larisa Miropolsky

`Get_Logistic_Weights` *Get the logistic weight*

Description

Get logistic weights from either a genotype matrix (Z) or a vector of minor allele frequncies (MAF). Users can apply this weights to SKAT by giving it as the "weights" parameter. The logistic weight gives equal weights to rare variants and nearly zero weight to common variants.

Usage

```
Get_Logistic_Weights(Z, par1=0.07, par2=150)


Get_Logistic_Weights_MAF(MAF, par1=0.07, par2=150)
```

Arguments

`Z`	a numeric genotype matrix with each row as a different individual and each column as a separate gene/snp. Each genotype should be coded as 0, 1, 2, and 9 for AA, Aa, aa, and missing, where A is a major allele and a is a minor allele.
`MAF`	a numeric vector of minor allele frequncies.
`par1`	a numeric value of the first parameter of the logistic weight (default= 0.07).
`par2`	a numeric value of the second parameter of the logistic weight(default= 150).

Details

The formula for the weight is

$$weights = \frac{e^{(par1-MAF)par2}}{1+e^{(par1-MAF)par2}}.$$

Value

A vector of the logistic weight.

Author(s)

Seunggeun Lee

Examples

```
data(SKAT.example)


############################################################
# Compute the P-value of SKAT with the logistic Weight (par1=0.07, par2=150)

# Use logistic weight
Z<-SKAT.example$Z
obj<-SKAT_Null_Model(y.c ~ X, out_type="C", data=SKAT.example)
weights<-Get_Logistic_Weights(Z, par1=0.07, par2=150)
SKAT(Z, obj, kernel = "linear.weighted", weights=weights)$p.value

# Weights function
MAF<-colMeans(Z)/2
plot(MAF,weights)
```

Get_RequiredSampleSize

Get the required sample size to achieve the given power

Description

Get the sample sizes required to achieve the given power.

Usage

```
Get_RequiredSampleSize(obj, Power=0.8)
```

Arguments

obj	an object returned from Power_Continuous or Power_Logistic.
Power	a value of the power to be achieved (default= 0.8).

Details

This function computes required sample sizes using simple interpolation.

Value

A list object for the required sample sizes.

Author(s)

Seunggeun Lee

`Get_Resampling_Pvalue` *Compute a resampling p-value*

Description

Compute a resampling p-value using resampled residuals. To use it, SKAT_Null_Model or SKAT_Null_Model_MomentAdju
should have n.Resampling > 0.

Usage

```
Get_Resampling_Pvalue(obj)

Get_Resampling_Pvalue_1(p.value, p.value.resampling)
```

Arguments

`obj`	SKAT outcome object.
`p.value`	a numeric value of the SKAT p-value.
`p.value.resampling`	a vector of p-values from the resampled residuals.

Details

See SKAT_Null_Model

Value

`p.value`	the resampling p-value. It is computed as (n1 +1)/(n+1), where n is the number of resampling (n.Resampling in SKAT_Null_Model or SKAT_Null_Model_MomentAdjust), and n1 is the number of resampled residual p-values smaller than the original sample p-value.
`is_smaller`	a logical value indicates whether the resampling p-value should be smaller. If n1=0, then it is TRUE, otherwise it is FALSE.

Author(s)

Seunggeun Lee

Open_SSD *Open SNP set data file (SSD)*

Description

Open a SNP Set data file (SSD). After finishing using the SSD file, you must close the file by callinsg Close_SSD function.

Usage

```
Open_SSD(File.SSD, File.Info)
```

Arguments

File.SSD	name of the SSD file .
File.Info	name of the SSD info file.

Value

a list object of SSD.INFO.

Author(s)

Seunggeun Lee, Larisa Miropolsky

Power_Continuous *Power calculation, continuous traits*

Description

Compute an average power of SKAT and SKAT-O for testing association between a genomic region and continuous phenotypes with a given disease model.

Usage

```
Power_Continuous(Haplotypes=NULL, SNP.Location=NULL, SubRegion.Length=-1
, Causal.Percent=5, Causal.MAF.Cutoff=0.03, alpha =c(0.01,10^(-3),10^(-6))
, N.Sample.ALL = 500 * (1:10), Weight.Param=c(1,25), N.Sim=100
, BetaType = "Log", MaxBeta=1.6, Negative.Percent=0)


Power_Continuous_R(Haplotypes=NULL, SNP.Location, SubRegion.Length=-1
, Causal.Percent=5, Causal.MAF.Cutoff=0.03, alpha =c(0.01,10^(-3),10^(-6))
, N.Sample.ALL = 500 * (1:10), Weight.Param=c(1,25), N.Sim=100
, BetaType = "Log", MaxBeta=1.6, Negative.Percent=0, r.corr=0)
```

Arguments

Haplotypes	a haplotype matrix with each row as a different individual and each column as a separate SNP (default= NULL). Each element of the matrix should be either 0 (major allel) or 1 (minor allele). If NULL, SKAT.haplotype dataset will be used to compute power.
SNP.Location	a numeric vector of SNP locations that should be matched with the SNPs in the Haplotype matrix (default= NULL). It is used to obtain subregions. When Haplotype=NULL, it should be NULL.
SubRegion.Length	a value of the length of subregions (default= -1). Each subregion will be randomly selected, and then the average power will be calculated by taking the average over the estimated powers of all subregions. If SubRegion.Length=-1 (default), the length of the subregion will be the same as the length of the whole region, so there will no random selection of subregions.
Causal.Percent	a value of the percentage of causal SNPs among rare SNPs (MAF < Causal.MAF.Cutoff)(default= 5).
Causal.MAF.Cutoff	a value of MAF cutoff for the causal SNPs. Only SNPs that have MAFs smaller than the cutoff will be considered as causal SNPs (default= 0.03).
alpha	a vector of the significance levels (default= c(0.01,10^(-3),10^(-6))).
N.Sample.ALL	a vector of the sample sizes (default= 500 * (1:10)).
Weight.Param	a vector of parameters of beta weights (default= c(1,25)).
N.Sim	a value of number of causal SNP/SubRegion sets to be generated to compute the average power (default= 100). Power will be computed for each causal SNP/SubRegion set, and then the average power will be obtained by taking average over the computed powers.
BetaType	a type of effect sizes (default= "Log"). "Log" indicates that effect sizes of causal variants equal to $c\|log10(MAF)\|$, and "Fixed" indicates that effect sizes of all causal variants are the same.
MaxBeta	a numeric value of the maximum effect size (default= 1.6). When BetaType="Log", the maximum effect size is MaxBeta (when MAF=0.0001). When BetaType="Fixed", all causal variants have the same effect size (= MaxBeta). See details
Negative.Percent	a numeric value of the percentage of coefficients of causal variants that are negative (default= 0).
r.corr	(Power_Continuous_R only) the ρ parameter for the compound symmetric correlation kernel (default= 0). See details.

Details

By default it uses the haplotype information in the SKAT.haplotypes dataset. So if you want to use the SKAT.haplotypes dataset, you can left Haplotypes and SNP.Location as NULL.

When BetaType="Log", MaxBeta is a coeffecient value (β) of the causal SNP at MAF $= 10^{-4}$ and used to obtain c value of the function $c|log10(MAF)|$. For example, if MaxBeta=1.6, $c = 1.6/4 = 0.4$. Then a variant with MAF=0.001 has $\beta = 1.2$ and a variant with MAF=0.01 has $\beta = 0.8$.

When SubRegion.Length is small such as 3kb or 5kb, it is possible that you can have different estimated power for each run with N.Sim = $50 \sim 100$. Then, please increase the N.Sim to $500 \sim 1000$ to obtain stable results.

R.sq is computed under the no linkage disequilibrium assumption.

Power_Continuous_R computes power with new class of kernels with the compound symmetric correlation structure. It uses a slightly different approach, and thus Power_Continuous and Power_Continuous_R can produce slightly different results although r.corr=0.

If you want to computer power of SKAT-O by estimating the optimal r.corr, use r.corr=2. The estimated optimal r.corr is $r.corr = p_1^2(2p_2 - 1)^2$, where p_1 is a proportion of causal variants, and p_2 is a proportion of negatively associated causal variants among the causal variants.

Value

`Power`	A matrix with each row as a different sample size and each column as a different significance level. Each element of the matrix is the estimated power.
`R.sq`	Proportion of phenotype variance explained by genetic variants.
`r.corr`	r.corr value. When r.corr=2 is used, it provides the estimated r.corr value. See details.

Author(s)

Seunggeun Lee

Examples

```
#
# Calculate the average power of randomly selected 3kb regions
# with the following conditions.
#
# Causal percent = 20%
# Negative percent = 20%
# Max effect size  = 2 at MAF = 10^-4
#
# When you use this function, please increase N.Sim (more than 100)
#

out.c<-Power_Continuous(SubRegion.Length=3000,
Causal.Percent= 20, N.Sim=5, MaxBeta=2,Negative.Percent=20)
out.c

#
# Calculate the required sample sizes to achieve 80% power

Get_RequiredSampleSize(out.c, Power=0.8)
```

Power_Logistic *Power calculation, Dichotomous traits*

Description

Compute an average power of SKAT and SKAT-O for testing association between a genomic region and dichotomous phenotypes from case-control studies with a given disease model.

Usage

```
Power_Logistic(Haplotypes = NULL, SNP.Location = NULL, SubRegion.Length=-1
, Prevalence=0.01, Case.Prop=0.5, Causal.Percent=5, Causal.MAF.Cutoff=0.03
, alpha =c(0.01,10^(-3),10^(-6)), N.Sample.ALL = 500 * (1:10)
, Weight.Param=c(1,25), N.Sim=100, OR.Type = "Log"
, MaxOR=5, Negative.Percent=0)

Power_Logistic_R(Haplotypes = NULL, SNP.Location = NULL, SubRegion.Length=-1
, Prevalence=0.01, Case.Prop=0.5, Causal.Percent=5, Causal.MAF.Cutoff=0.03
, alpha =c(0.01,10^(-3),10^(-6)), N.Sample.ALL = 500 * (1:10)
, Weight.Param=c(1,25), N.Sim=100, OR.Type = "Log"
, MaxOR=5, Negative.Percent=0, r.corr=0)
```

Arguments

Haplotypes	a haplotype matrix with each row as a different individual and each column as a separate SNP (default= NULL). Each element of the matrix should be either 0 (major allel) or 1 (minor allele). If NULL, SKAT.haplotype dataset will be used to compute power.
SNP.Location	a numeric vector of SNP locations which should be matched with the SNPs in the Haplotype matrix (default= NULL). It is used to obtain subregions. When Haplotype=NULL, it should be NULL.
SubRegion.Length	a value of the length of subregions (default= -1). Each subregion will be randomly selected, and then the average power will be calculated by taking the average over the estimated powers of all subregions. If SubRegion.Length=-1 (default), the length of the subregion is the same as the length of the whole region, so there will no random selection of subregions.
Prevalence	a value of disease prevalence.
Case.Prop	a value of the proportion of cases. For example, Case.Prop=0.5 means 50 % of samples are cases and 50 % of samples are controls.
Causal.Percent	a value of the percentage of causal SNPs among rare SNPs (MAF < Causal.MAF.Cutoff)(default= 5).

`Causal.MAF.Cutoff`
: a value of MAF cutoff for the causal SNPs. Only SNPs that have MAFs smaller than this are considered as causal SNPs (default= 0.03).

`alpha`
: a vector of the significance levels (default= c(0.01,10^(-3),10^(-6))).

`N.Sample.ALL`
: a vector of the sample sizes (default= 500 * (1:10)).

`Weight.Param`
: a vector of parameters of beta weights (default= c(1,25)).

`N.Sim`
: a value of number of causal SNP/SubRegion sets to be generated to compute the average power (default= 100). Power will be computed for each causal SNP/SubRegion set, and then the average power will be obtained by taking mean of the computed powers.

`OR.Type`
: a function type of effect sizes (default= "Log"). "Log" indicates that log odds ratio of causal variants equal to $c|log10(MAF)|$, and "Fixed" indicates that log odds ratio of all causal variants are the same.

`MaxOR`
: a numeric value of the maximum odds ratio (default= 5). When OR.Type="Log", the maximum odds ratio is MaxOR (when MAF=0.0001). When OR.Type="Fixed", all causal variants have the same odds ratio (= MaxOR). See details

`Negative.Percent`
: a numeric value of the percentage of coefficients of causal variants that are negative (default= 0).

`r.corr`
: (Power_Logistic_R only) the ρ parameter of new class of kernels with compound symmetric correlation structure for genotype effects (default= 0). See details.

Details

By default it uses the haplotype information in the SKAT.haplotypes dataset. So you can left Haplotypes and SNP.Location as NULL if you want to use the SKAT.haplotypes dataset.

When OR.Type="Log", MaxOR is a odds ratio of the causal SNP at MAF $= 10^{-4}$ and used to obtain c value in the function $\log OR = c|log10(MAF)|$. For example, if MaxOR=5, $c = log(5)/4 = 0.402$. Then a variant with MAF=0.001 has log odds ratio = 1.206 and a variant with MAF=0.01 has log odds ratio = 0.804.

When SubRegion.Length is small such as 3kb or 5kb, it is possible that you can have different estimated power for each run with N.Sim = $50 \sim 100$. Then, please increase N.Sim to $500 \sim 1000$ to obtain stable results.

Power_Logistic_R computes the power with new class of kernels with the compound symmetric correlation structure. It uses a slightly different approach, and thus Power_Logistic and Power_Logistic_R can produce slightly different results although r.corr=0.

If you want to computer power of SKAT-O by estimating the optimal r.corr, use r.corr=2. The estimated optimal r.corr is $r.corr = p_1^2(2p_2 - 1)^2$, where p_1 is a proportion of causal variants, and p_2 is a proportion of negatively associated causal variants among the causal variants.

Value

`Power`
: A matrix with each row as a different sample size and each column as a different significance level. Each element of the matrix is the estimated power.

`r.corr`
: r.corr value. When r.corr=2 is used, it provides the estimated r.corr value. See details.

Author(s)

Seunggeun Lee

Examples

```
#
# Calculate the average power of randomly selected 3kb regions
# with the following conditions.
#
# Causal percent = 20%
# Negative percent = 20%
# Max OR = 7 at MAF = 10^-4
#
# When you use this function, please increase N.Sim (more than 100)
#
out.b<-Power_Logistic(SubRegion.Length=3000,
Causal.Percent= 20, N.Sim=5 ,MaxOR=7,Negative.Percent=20)

out.b

#
# Calculate the required sample sizes to achieve 80% power

Get_RequiredSampleSize(out.b, Power=0.8)
```

QQPlot_Adj *Adjusted QQ plot*

Description

Draws a MAP-adjusted QQ plot

Usage

```
QQPlot_Adj(Pval, MAP, main="QQ plot", ntry=500, confidence=0.95, Is.unadjsted=TRUE
, Is.legend=TRUE, xlab="Expected Quantiles (-log10 P-values)"
, ylab="Observed Quantiles (-log10 P-values)")
```

Arguments

Pval	a vector of p-values.
MAP	a vector of MAP.
main	a main title.
ntry	a numeric value of the number for resampling (default= 500).
confidence	a value for the confidence band (default= 0.95).
Is.unadjsted	a logical value to indicate whether to plot the unadjusted QQ plot (default= TRUE).
Is.legend	a logical value to indicate whether to plot a legend (default= TRUE).
xlab	a label for the x axis.
ylab	a label for the y axis.

Author(s)

Seunggeun Lee

References

Lee, S., Fuchsberger, C., Kim, S., Scott, L. (2015) An efficient resampling method for calibrating single and gene-based rare variant association analysis in case-control studies. *Biostatistics*, in press.

Read_Plink_FAM *Read Plink FAM and covariates files*

Description

Read Plink FAM and covariates files.

Usage

```
Read_Plink_FAM(Filename, Is.binary=TRUE, flag1=0)
Read_Plink_FAM_Cov(Filename, File_Cov, Is.binary=TRUE, flag1=0, cov_header=TRUE)
```

Arguments

Filename	input file name of plink FAM file
Is.binary	if TRUE, the phenotype is binary. If phenotype is continuous, it should be FALSE
flag1	0 represents the default coding of unaffected/affected (1/2) (default=0), and 1 represents 0/1 coding. flag1=1 is the same as –1 flag in plink. Please see the plink manual.

File_Cov	an input file name of plink covariates file. The first two columns of this file should be FID and IID.
cov_header	a logical value indicating whether the covariate file contains a header row (default=TRUE)

Value

A dataframe of Family ID (FID), Individual ID (IID), Paternal ID (PID), Maternal ID(MID), Sex, and Phenotype. If Read_Plink_FAM_Cov is used with a covariate file, the dataframe has covariates from the 7th column.

Author(s)

Seunggeun Lee

Read_SNP_WeightFile *Read a file with custom weights*

Description

Read a file with custom weights

Usage

```
Read_SNP_WeightFile(FileName)
```

Arguments

FileName	input file name of a custom weight.

Details

The file should be a white-space (space or tab) delimitered file with 2 columns: SNP_ID and weight value.

Please keep in mind that there should be no header!!

Value

Output object has a hash table of SNP IDs and weights.

Author(s)

Seunggeun Lee

`Resampling_FWER` *Obtain significant SNP sets after controlling family wise error rate (FWER)*

Description

Obtain significant SNP sets after controlling for family wise error rate (FWER) using resampled residuals. To use it, SKAT_Null_Model or SKAT_Null_Model_MomentAdjust should have n.Resampling > 0.

Usage

```
Resampling_FWER(obj,FWER=0.05)

Resampling_FWER_1(P.value, P.value.Resampling, FWER=0.05)
```

Arguments

`obj`	object returned from SKAT.SSD.All or SKATBinary.SSD.All.
`P.value`	a vector of SKAT p-values. If 100 genes were tested, this vector should have 100 p-values.
`P.value.Resampling`	a matrix of p-values of the resampled residuals. Each row represents each gene/snp set, and each column represents resampling set. For example, if you have 100 genes, and conducted resampling 1000 times (ex.n.Resampling=1000 in SKAT_Null_Model), then it should be a 100 x 1000 matrix.
`FWER`	a numeric value of FWER rate to control (default=0.05)

Value

`results`	If the returned object from SKAT.SSD.all (or SKATBinary.SSD.All) are used, it is a sub-table of significant snp sets of the result table in the obj. If you use P.value and P.value.Resampling, it is a vector of significant p-values. If there is no significant snp set, it is NULL.
`n`	a numeric value of the number of significant snp sets.
`ID`	a vector of indexes of significant snp sets.

Author(s)

Seunggeun Lee

SKAT *SNP-set (Sequence) Kernel Association Test*

Description

Test for association between a set of SNPS/genes and continuous or dichotomous phenotypes using kernel regression framework.

Usage

```
SKAT(Z, obj, kernel = "linear.weighted",
  method="davies", weights.beta=c(1,25), weights=NULL,
  impute.method="fixed", r.corr=0, is_check_genotype=TRUE,
  is_dosage = FALSE, missing_cutoff=0.15 , max_maf=1, estimate_MAF=1)

SKAT.SSD.OneSet(SSD.INFO, SetID, obj, ... ,obj.SNPWeight=NULL)

SKAT.SSD.OneSet_SetIndex(SSD.INFO, SetIndex, obj, ... ,obj.SNPWeight=NULL)
```

Arguments

Z	a numeric genotype matrix with each row as a different individual and each column as a separate gene/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.
obj	an output object of the SKAT_Null_Model function.
kernel	a type of kernel (default= "linear.weighted"). See detail section.
method	a method to compute the p-value (default= "davies"). "davies" represents an exact method that computes the p-value by inverting the characteristic function of the mixture chisq, "liu" represents an approximation method that matches the first 3 moments, "liu.mod" represents modified "liu" method that matches kurtosis instead of skewness to improve tail probability approximation,"SKATO" and "optimal.adj" represent a SKAT-O based on an unified approach, and "optimal" is an old version of the implementation of SKAT-O. See details.
weights.beta	a numeric vector of parameters for the beta weights for the weighted kernels. If you want to use your own weights, please use the "weights" parameter. It will be ignored if "weights" parameter is not null.
weights	a numeric vector of weights for the weighted kernels. It is $\sqrt{w}$ in the SKAT paper. So if you want to use the Madsen and Browning (2009) weight, you should set each element of weights as $1/\sqrt{p(1-p)}$, not $1/p(1-p)$. When it is NULL, the beta weight with the "weights.beta" parameter is used.

`impute.method` a method to impute missing genotypes (default= "fixed"). "bestguess" imputes missing genotypes as most likely values (0,1,2), "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 values (2p).

`r.corr` the ρ parameter for the compound symmetric correlation structure kernels (default= 0). If you give a vector value, SKAT will conduct the optimal test. It will be ignored if method="optimal" or method="optimal.adj". See details.

`is_check_genotype` a logical value indicating whether to check the validity of the genotype matrix Z (default= TRUE). If Z has non-SNP data, please set it FALSE, otherwise you will get an error message. If it is FALSE and you use weighted kernels, the weights should be given through the "weights" parameter.

`is_dosage` a logical value indicating whether the matrix Z is a dosage matrix. If it is TRUE, SKAT will ignore "is_check_genotype".

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

`max_maf` a cutoff of the maximum minor allele frequencies (MAF) (default=1, no cutoff). Any SNPs with MAF > cutoff will be excluded from the analysis.

`estimate_MAF` a numeric value indicating how to estimate MAFs for the weight calculation and the missing genotype imputation. If estimate_MAF=1 (default), SKAT uses all samples to estimate MAFs. If estimate_MAF=2, only samples with non-missing phenotypes and covariates are used to estimate MAFs.

`SSD.INFO` an SSD_INFO object returned from Open_SSD.

`SetID` a character value of Set ID. A set ID of each set can be found from SetInfo object in SSD.INFO.

`SetIndex` a numeric value of Set index. A set index of each set can be found from SetInfo object in SSD.INFO.

`...` further arguments to be passed to "SKAT"

`obj.SNPWeight` an output object of Read_SNP_WeightFile (default=NULL). If NULL, the beta weight with the "weights.beta" parameter will be used.

Details

There are 6 types of pre-specified kernels: "linear", "linear.weighted", "IBS", "IBS.weighted", "quadratic" and "2wayIX". Among them, "2wayIX" is a product kernel consisting of main effects and SNP-SNP interaction terms.

If users want to use dosage values instead of genotypes, set is_dosage=TRUE. Please keep in mind that plink formatted files (so SSD files) cannot used for dosages. Instead, you should make a genotype matrix Z to run SKAT.

The kernel matrix for the weighted linear kernel is $K = GWWG$, where G is a genotype matrix and W is a diagonal weight matrix. Please note that it is different from the notation we used in the original SKAT paper, which was $K = GWG$. The Madsen and Browning (2009) weight is $w = 1/\sqrt{p(1-p)}$ in the current notation. By the previous notation, it is $w = 1/p(1-p)$.

If you want to use the SSD file, you need to open it first using Open_SSD, and then use either SKAT.SSD.OneSet or SKAT.SSD.OneSet_SetIndex. Set index is a numeric value and automatically assigned to each set (from 1).

The r.corr represents a ρ parameter of the unified test, $Q_\rho = (1-\rho)Q_S + \rho Q_B$, where Q_S is a SKAT test statistic, and Q_B is a weighted burden test statistic. Therefore, $\rho = 0$ results in the original weighted linear kernel SKAT, and $\rho = 1$ results in the weighted burden test (default: $\rho = 0$). If r.corr is a vector, SKAT-O will be conducted with adaptively seleting ρ from given r.corr values. ρ should be a value between 0 and 1. When method="optimal" or method="optimal.adj", the r.corr parameter will be ignored.

We slightly changed the implementation for SKAT-O to improve the estimation of p-values. You can run it by using method="optimal.adj" or "SKATO". It uses a grid of eight points $\rho = (0, 0.1^2, 0.2^2, 0.3^2, 0.4^2, 0.5^2, 0.5, 1)$ for the search of the optimal ρ. If you want to use the original implementation of SKAT-O, use method="optimal" that carries out SKAT-O with an equal sized grid of 11 points (from 0 to 1).

If the true p.value is very small, you can have p.value=0 due to numerical reasons. In this case, please see pval.zero.msg that shows how small it is. For example, if the p.value is smaller than 10^{-60}, it has "Pvalue < 1.000000e-60".

By default, SKAT uses impute.method="fixed" that imputes missing genotypes as the mean genotype values (2p). When variates are very rare and missing rates between cases and controls are highly unbalanced, impute.method="fixed" can yield inflated type I error rate. In this case, we recommend to use impute.method="bestguess", which does not suffer the same problem.

Value

`p.value` p-value of SKAT.

`p.value.resampling`
p-values from resampled outcomes. You can get it when you use obj from SKAT_Null_Model function with resampling. See the SKAT_Null_Model.

`p.value.noadj` p-value of SKAT without the small sample adjustment. It only appears when small sample adjustment is applied.

`p.value.noadj.resampling`
p-values from resampled outcomes without the small sample adjustment.

`pval.zero.msg` (only when p.value=0) text message that shows how small the p.value is. ex. "Pvalue < 1.000000e-60" when the p.value is smaller than 10^{-60}

`Q` test statistic of SKAT. It has NA when method="optimal.adj" or "optimal".

`param` estimated parameters of each method.

`param$Is_Converged`
(only with method="davies") an indicator of the convergence (1=convergence, 0=non-convergence). When 0 (not converged), "liu" method will be used to compute p-values.

`param$n.marker` a number of SNPs in the genotype matrix.

`param$n.marker.test`
a number of SNPs used for the test. It can be different from param$n.marker when some markers are monomorphic or have higher missing rates than the missing_cutoff.

`test.snp.mac` a vector of minor allele count (MAC) of the snps tested. The name is SNP-ID.

Author(s)

Seunggeun Lee, Micheal Wu

References

Lee, S., Emond, M.J., Bamshad, M.J., Barnes, K.C., Rieder, M.J., Nickerson, D.A., NHLBI GO Exome Sequencing Project-ESP Lung Project Team, Christiani, D.C., Wurfel, M.M. and Lin, X. (2012) Optimal unified approach for rare variant association testing with application to small sample case-control whole-exome sequencing studies. *American Journal of Human Genetics*, 91, 224-237.

Lee, S., Wu, M. C., and Lin, X. (2012) Optimal tests for rare variant effects in sequencing association studies. *Biostatistics*, 13, 762-775.

Wu, M. C.*, Lee, S.*, Cai, T., Li, Y., Boehnke, M., and Lin, X. (2011) Rare Variant Association Testing for Sequencing Data Using the Sequence Kernel Association Test (SKAT). *American Journal of Human Genetics*, 89, 82-93. \ * contributed equally.

Wu, M. C., Kraft, P., Epstein, M. P.,Taylor, D., M., Chanock, S. J., Hunter, D., J., and Lin, X. (2010) Powerful SNP Set Analysis for Case-Control Genome-wide Association Studies. *American Journal of Human Genetics*, 86, 929-942.

Davies R.B. (1980) Algorithm AS 155: The Distribution of a Linear Combination of chi-2 Random Variables, *Journal of the Royal Statistical Society. Series C* , 29, 323-333.

H. Liu, Y. Tang, H.H. Zhang (2009) A new chi-square approximation to the distribution of non-negative definite quadratic forms in non-central normal variables, *Computational Statistics and Data Analysis*, 53, 853-856.

Duchesne, P. and Lafaye De Micheaux, P. (2010) Computing the distribution of quadratic forms: Further comparisons between the Liu-Tang-Zhang approximation and exact methods, *Computational Statistics and Data Analysis*, 54, 858-862.

Examples

```
data(SKAT.example)


############################################################
# SKAT with default Beta(1,25) Weights
# - without covariates
Z<-SKAT.example$Z
# continuous trait
obj<-SKAT_Null_Model(y.c ~ 1, out_type="C", data=SKAT.example)
SKAT(Z, obj)$p.value

# dichotomous trait
obj<-SKAT_Null_Model(y.b ~ 1, out_type="D", data=SKAT.example)
SKAT(Z, obj)$p.value


################################################
# SKAT with default Beta(1,25) Weights
# - with covariates
```

```
# continuous trait
obj<-SKAT_Null_Model(y.c ~ X, out_type="C", data=SKAT.example)
SKAT(Z, obj)$p.value

obj.b<-SKAT_Null_Model(y.b ~ X, out_type="D", data=SKAT.example)
SKAT(Z, obj.b)$p.value

################################################
# SKAT with default Beta(1,25) Weights
# - Optimal Test

SKAT(Z, obj, method="optimal.adj")$p.value

# you can get the same p-value by using method="SKATO"
SKAT(Z, obj, method="SKATO")$p.value

############################################################
# SKAT with Beta(1,30) Weights

SKAT(Z, obj, weights.beta=c(1,30))$p.value
```

SKAT.example *Example data for SKAT*

Description

Example data for SKAT.

Format

SKAT.example contains the following objects:

Z a numeric genotype matrix of 2000 individuals and 67 SNPs. Each row represents a different individual, and each column represents a different SNP marker.

X a numeric matrix of 2 covariates.

y.c a numeric vector of continuous phenotypes.

y.b a numeric vector of binary phenotypes.

SKAT.example.ChrX *Example data for SKAT*

Description

Example data for SKAT.

Format

SKAT.example contains the following objects:

Z a numeric genotype matrix of 2000 individuals and 52 SNPs. Each row represents a different individual, and each column represents a different SNP marker.

x1 a numeric vector of continuous covariates.

x2 a numeric vector of binary covariates.

Gender a numeric vector of gender (male=1, female=2).

y a numeric vector of binary phenotypes.

Z.A a list object of 10 genotype matrices.

SKAT.fam.example *Example data for SKAT_NULL_emmaX*

Description

Example data for SKAT_emmaX.

Format

Example contains the following objects:

Z a numeric genotype matrix Each row represents a different individual, and each column represents a different SNP marker.

X a numeric matrix of covariates.

y a numeric vector of continuous phenotypes.

K a kinship matrix.

SKAT.haplotypes *Haplotype dataset for power calculation*

Description

Haplotype dataset generated by COSI with mimicking linkage disequilibrium (LD) structure of European ancestry.

Format

This list object contains the following objects:

Haplotype a numeric matrix of 10,000 haplotypes over 200k BP region. Each row represents a different haplotype, and each column represents a different SNP marker. It is generated by the calibration coalescent model (COSI) with mimicking LD structure of European ancestry.

SNPInfo a dataframe of SNP information.

References

Schaffner, S.F. and Foo, C. and Gabriel, S. and Reich, D. and Daly, M.J. and Altshuler, D. (2005) Calibrating a coalescent simulation of human genome sequence variation. *Genome Research*, 15, 1576-1583.

SKAT.SSD.All *SNP-set Kernel Association Test*

Description

Iteratively carry out association tests with phenotypes and SNP sets in SSD file.

Usage

```
SKAT.SSD.All(SSD.INFO, obj, ..., obj.SNPWeight=NULL)

SKATBinary.SSD.All(SSD.INFO, obj, ..., obj.SNPWeight=NULL)

SKATBinary_Robust.SSD.All(SSD.INFO, obj, ...,obj.SNPWeight=NULL)

SKAT_CommonRare.SSD.All(SSD.INFO, obj, ..., obj.SNPWeight=NULL)

SKAT_CommonRare_Robust.SSD.All(SSD.INFO, obj, ..., obj.SNPWeight=NULL)
```

Arguments

SSD.INFO	SSD_INFO object returned from Open_SSD.
obj	output object from SKAT_Null_Model.
...	further arguments to be passed to "SKAT" or "SKATBinary".
obj.SNPWeight	output object from Read_SNP_WeightFile (default=NULL). If NULL, the beta weight with the "weights.beta" parameter will be used.

Details

Please see SKAT or SKATBinary for details.

Value

results	dataframe that contains SetID, p-values (P.value), the number of markers in the SNP sets (N.Marker.All), and the number of markers to test for an association after excluding non-polymorphic or high missing rates markers (N.Marker.Test). The output dataframe from SKATBinary.SSD.All (and others) have more columns. For example, the outcome from SKATBinary.SSD.All have columns for the method to compute p-values and the minimum achievable p-values (MAP).
P.value.Resampling	the matrix that contains p-values of resampled phenotypes.
OUT.snp.mac	each element in the list is a vector of MAC of SNPs used in the test. The names are SNP-IDs.

Author(s)

Seunggeun Lee

SKATBinary *SNP set test for binary traits with asymptotic and efficient resampling methods*

Description

This function computes p-values of Burden test, SKAT, and SKAT-O for binary traits using asymptotic and efficient resampling methods.

Usage

```
SKATBinary(Z, obj, kernel = "linear.weighted", method="SKAT"
, method.bin="Hybrid", weights.beta=c(1,25), weights = NULL
, r.corr=0, impute.method = "bestguess", is_check_genotype=TRUE
, is_dosage = FALSE, missing_cutoff=0.15, max_maf=1
, estimate_MAF=1, N.Resampling=2 *10^6, seednum=100, epsilon=10^-6
, SetID=NULL)
```

```
SKATBinary.SSD.OneSet(SSD.INFO, SetID, obj, ... ,obj.SNPWeight=NULL)

SKATBinary.SSD.OneSet_SetIndex(SSD.INFO, SetIndex, obj, ... ,obj.SNPWeight=NULL)
```

Arguments

Z	a numeric genotype matrix with each row as a different individual and each column as a separate gene/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.
obj	output object from SKAT_Null_Model.
kernel	type of kernel (default= "linear.weighted"). See SKAT page for details.
method	type of gene based test (default= "SKAT"). The possible choices are "SKAT", "Burden" and "SKATO", which represents SKAT, Burden and SKAT-O tests, respectively. This parameter differs from the "method" parameter in SKAT function. "Burden" is equivalent to method="davies" and r.corr=1 and "SKATO" is equivalent to method="optimal.adj" in the SKAT function. When method="Burden" or method="SKATO", r.corr will be ignored.
method.bin	type of method to compute a p-value (default="Hybrid"). Possible choices are "Hybrid", "ER", "ER.A", "QA", "MA" and "UA". See details
weights.beta	a numeric vector of parameters of beta weights. It is only used for weighted kernels. If you want to use your own weights, please specify the "weights" parameter.
weights	a numeric vector of weights for the weighted kernels. See SKAT page for details.
impute.method	a method to impute missing genotypes (default= "bestguess"). "bestguess" imputes missing genotypes as most likely values (0,1,2), "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).
r.corr	the ρ parameter for the compound symmetric correlation structure kernel (default= 0). If it is a vector, SKAT will conduct the optimal test. It is ignored when method="Burden" or method="SKATO".
is_check_genotype	a logical value indicating whether to check the validity of the genotype matrix Z (default= TRUE). See SKAT page for details.
is_dosage	a logical value indicating whether the matrix Z is a dosage matrix. If it is TRUE, SKAT will ignore "is_check_genotype".
missing_cutoff	a cutoff of the missing rates of SNPs (default=0.15). Any SNPs with missing rates higher than the cutoff will be excluded from the analysis.
max_maf	a cutoff of the maximum minor allele frequencies (MAF) (default=1, no cutoff). Any SNPs with MAF > cutoff will be excluded from the analysis.
estimate_MAF	a numeric value indicating how to estimate MAFs for the weight calculation and the missing genotype imputation. See SKAT page for details.

N.Resampling	a number of resampling to be conducted to get p-values (default=2 *10^6).
seednum	a seed number for random number generation (default=100). If NULL, no seed number will be assigned.
epsilon	a precision level (default=10^-6).
SSD.INFO	an SSD_INFO object returned from Open_SSD.
SetID	a character value of Set ID. You can find a set ID of each set from SetInfo object of SSD.INFO. In SKATBinary function, this parameter is for the internal use only.
SetIndex	a numeric value of Set index. You can find a set index of each set from SetInfo object of SSD.INFO
...	further arguments to be passed to "SKATBinary"
obj.SNPWeight	an output object of Read_SNP_WeightFile (default=NULL). If NULL, the beta weight with the "weights.beta" parameter will be used.

Details

This function implements six methods (method.bin) to compute p-values: 1) Efficient resampling (ER); 2) Quantile adjusted moment matching (QA); 3) Moment matching adjustment (MA); 4) No adjustment (UA); 5) Adaptive ER (ER.A); and 6) Hybrid. "Hybrid" selects a method based on the total minor allele count (MAC), the number of individuals with minor alleles (m), and the degree of case-control imbalance. When method.bin="ER" or "ER.A", SKATBinary compute mid-p-values and minimum achievable mid p-values.

If seednum is not NULL, set.seed(seednum) function is used to specify seeds to get the same p-values of ER based methods for different runs. Therefore, please set seednum=NULL, if you do not want to set seeds.

SKATBinary uses impute.method="bestguess" as a default method for the imputation, which is different from SKAT that uses impute.method="fixed" as a default method. We changed it because SKATBinary with impute.method="fixed" can yield false positives when variates are very rare and missing rates between cases and controls are unbalanced. When missing rates between cases and controls are highly unbalanced, SKAT impute.method="fixed" can also yield false positives, but it happens less likely. So we did not change the default imputation method in SKAT.

Value

p.value	p-value. It will be the mid p-value if ER or ER.A are used to compute the p-value.
p.value.standard	(ER and ER.A only) standard p-value.
p.value.resampling	p-values from resampled outcome. You can obtain it when n.Resampling (in SKAT_Null_Model) > 0. See SKAT_Null_Model page.
p.value.standard.resampling	(ER and ER.A only)standard p-values from resampled outcomes.
m	the number of individuals with minor alleles.
MAP	minimum possible p-values. It is available when the method.bin="ER" and m is sufficiently small.

MAC	total minor allele count (MAC).
n.total	(ER only) the number of resampling to be generated to get the p-value. It can be smaller than N.Resampling when the total number of configurations of case-controls among individuals with minor alleles are smaller than N.Resampling.
is.accurate	logical value for the accuracy of the p-value. If it is false, more resampling is needed to accurately estimate the p-value.
param$n.marker	a number of SNPs in the genotype matrix
param$n.marker.test	a number of SNPs used for the test. It can be different from param$n.marker when some markers are monomorphic or have higher missing rates than the missing_cutoff.
method.bin	a type of method to be used to compute the p-value.
test.snp.mac	a vector of minor allele count (MAC) of the snps tested. The name is SNP-ID.

Author(s)

Seunggeun Lee

References

Lee, S., Fuchsberger, C., Kim, S., Scott, L. (2015) An efficient resampling method for calibrating single and gene-based rare variant association analysis in case-control studies. *Biostatistics*, in press.

Examples

```
data(SKATBinary.example)
Z<-SKATBinary.example$Z

obj<-SKAT_Null_Model(y ~ x1 + x2, out_type="D", data=SKATBinary.example)

# run SKAT (default method) with Hybrid
out = SKATBinary(Z, obj)

# p-value
out$p.value

# MAP
out$MAP

# method used to compute p-value (method.bin)
out$method.bin


#
# Run burden and SKAT-O with Hybrid

SKATBinary(Z, obj, method="Burden")$p.value
```

```
SKATBinary(Z, obj, method="SKATO")$p.value

#
# Run with SKAT-QA, -MA and -UA

SKATBinary(Z, obj, method.bin="QA")$p.value

SKATBinary(Z, obj, method.bin="MA")$p.value

SKATBinary(Z, obj, method.bin="UA")$p.value

# UA from SKAT function
SKAT(Z, obj)$p.value


#
# Run with Adaptive ER

out =SKATBinary(Z, obj, method.bin="ER.A")

out$p.value

# the number of total resampling is smaller than 2*10^6 (default value)
out$n.total
```

SKATBinary.example *Example data for SKAT*

Description

Example data for SKAT.

Format

SKAT.example contains the following objects:

Z a numeric genotype matrix of 2000 individuals and 11 SNPs. Each row represents a different individual, and each column represents a different SNP marker.

x1 a numeric vector of continuous covariates.

x2 a numeric vector of binary covariates.

y a numeric vector of binary phenotypes.

Z.A a list object of 30 genotype matrices.

SKATBinary_Robust *SNP set test for binary traits with robust region-based methods*

Description

This function computes p-values of robust burden test, SKAT, and SKAT-O for binary traits using SPA and ER.

Usage

```
SKATBinary_Robust(Z, obj, kernel = "linear.weighted", method="SKAT"
, r.corr=NULL, weights.beta=c(1,25), weights = NULL
, impute.method = "bestguess",  is_check_genotype=TRUE
  , is_dosage = FALSE, missing_cutoff=0.15, max_maf=1
, estimate_MAF=1)

SKATBinary_Robust.SSD.OneSet(SSD.INFO
, SetID, obj, ...,obj.SNPWeight=NULL)

SKATBinary_Robust.SSD.OneSet_SetIndex(SSD.INFO
, SetIndex, obj, ... ,obj.SNPWeight=NULL)
```

Arguments

Z	a numeric genotype matrix with each row as a different individual and each column as a separate gene/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. Now we support both matrix and sparse matrix.
obj	output object from SKAT_Null_Model.
kernel	type of kernel (default= "linear.weighted"). The possible choices are "linear" and "linear.weighted".
method	type of gene based test (default= "SKAT"). The possible choices are "SKAT", "Burden" and "SKATO", which represents robust SKAT, Burden and SKAT-O tests, respectively.
r.corr	the ρ parameter for all variants. ρ =0 and 1 indicate SKAT and Burden test, respectively.
weights.beta	a numeric vector of parameters of beta weights. It is only used for weighted kernels. If you want to use your own weights, please specify the "weights" parameter.
weights	a numeric vector of weights for the weighted kernels. See SKAT page for details.

`impute.method` a method to impute missing genotypes (default= "bestguess"). "bestguess" imputes missing genotypes as most likely values (0,1,2), "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).

`is_check_genotype` a logical value indicating whether to check the validity of the genotype matrix Z (default= TRUE). See SKAT page for details.

`is_dosage` a logical value indicating whether the matrix Z is a dosage matrix. If it is TRUE, SKAT will ignore "is_check_genotype".

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

`max_maf` a cutoff of the maximum minor allele frequencies (MAF) (default=1, no cutoff). Any SNPs with MAF > cutoff will be excluded from the analysis.

`estimate_MAF` a numeric value indicating how to estimate MAFs for the weight calculation and the missing genotype imputation. See SKAT page for details.

`SSD.INFO` an SSD_INFO object returned from Open_SSD.

`SetID` a character value of Set ID. You can find a set ID of each set from SetInfo object of SSD.INFO. In SKATBinary_Robust function, this parameter is for the internal use only.

`SetIndex` a numeric value of Set index. You can find a set index of each set from SetInfo object of SSD.INFO

`obj.SNPWeight` output object from Read_SNP_WeightFile (default=NULL). If NULL, the beta weight with the "weights.beta" parameter will be used.

`...` further arguments to be passed to "SKATBinary_Robust"

Value

`p.value` p-value. It will be the p-value based on robust methods.

`p.value_singlevariant` p-value for each single variant in this region-based test.

`mac` total minor allele count (MAC).

`param$n.marker` a number of SNPs in the genotype matrix.

`param$n.marker.test` a number of SNPs used for the test. It can be different from param$n.marker when some markers are monomorphic or have higher missing rates than the missing_cutoff.

`param$rho` the ρ parameter for all variants.

`test.snp.mac` a vector of minor allele count (MAC) of the snps tested. The name is SNP-ID.

Author(s)

Zhangchen Zhao

References

Zhao, Z., Bi, W., Zhou, W., VandeHaar, P., Fritsche, L. G., & Lee, S. (2019). UK-Biobank Whole Exome Sequence Binary Phenome Analysis with Robust Region-based Rare Variant Test. *The American Journal of Human Genetics*, in press.

Examples

```
data(SKATBinary.example)
Z<-SKATBinary.example$Z

obj<-SKAT_Null_Model(y ~ x1 + x2, out_type="D", data=SKATBinary.example)

# run SKAT (default method) with Hybrid
out = SKATBinary_Robust(Z, obj)

# p-value
out$p.value

#
# Run burden and SKAT

SKATBinary_Robust(Z, obj, method="Burden")$p.value
SKATBinary_Robust(Z, obj, method="SKAT")$p.value
```

`SKATBinary_Single` *Single variant tests for binary traits with Firth and efficient resampling methods*

Description

This function computes p-values of single variant test using the firth and efficient resampling methods.

Usage

```
SKATBinary_Single(Z, obj, method.bin="Hybrid"
, impute.method = "bestguess", is_check_genotype=TRUE, is_dosage = FALSE
, missing_cutoff=0.15, max_maf=1, estimate_MAF=1
, N.Resampling=2*10^6, seednum=100, epsilon=10^-6)
```

Arguments

Z	a numeric genotype vector. 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.
obj	output object from SKAT_Null_Model.
method.bin	a type of method to compute a p-value (default="Hybrid"). See details.
impute.method	a method to impute missing genotypes (default= "bestguess").
is_check_genotype	a logical value indicating whether to check the validity of the genotype matrix Z (default= TRUE). See SKAT page for details.
is_dosage	a logical value indicating whether the matrix Z is a dosage matrix. If it is TRUE, SKAT will ignore "is_check_genotype".
missing_cutoff	a cutoff of the missing rates of SNPs (default=0.15). Any SNPs with missing rates higher than the cutoff will be excluded from the analysis.
max_maf	a cutoff of the maximum minor allele frequencies (MAF) (default=1, no cutoff). Any SNPs with MAF > cutoff will be excluded from the analysis.
estimate_MAF	a numeric value indicating how to estimate MAFs for the weight calculation and the missing genotype imputation. See SKAT page for details
N.Resampling	a number of resampling to be conducted to get p-values (default=2 *10^6).
seednum	a seed number for random number generation (default=100). If NULL, no seed number will be assigned.
epsilon	a precision level (default=10^-6).

Details

This function implements three methods (method.bin) to compute p-values: 1) Efficient resampling (ER); 2) Firth biased adjusted likelihood ratio test (Firth); and 3) Hybrid. "Hybrid" selects a method based on the total minor allele count (MAC), the number of individuals with minor alleles (m), and the degree of case-control imbalance.

Adaptive ER (ER.A) is not implemented yet.

If seednum is not NULL, set.seed(seednum) function is used to specify seeds to get the same p-values of ER based methods for different runs. Therefore, please set seednum=NULL, if you do not want to set seeds.

Value

p.value	p-value. It will be the mid p-value if ER is used to compute the p-value.
p.value.standard	(ER only) standard p-value.
p.value.resampling	p-values from resampled outcome. You can obtain it when n.Resampling in SKAT_Null_Model was > 0. See the SKAT_Null_Model.
p.value.standard.resampling	(ER only) standard p-values from resampled outcome.

m	the number of individuals with minor alleles.
MAP	the minimum possible p-values. It is available when the method.bin="ER" and m is sufficiently small.
MAC	the total minor allele count (MAC).
n.total	(ER only) the number of resampling to be generated to get the p-value. It can be smaller than N.Resampling when the total number of configurations of case-controls among individuals with minor alleles are smaller than N.Resampling.
is.accurate	logical value for the accuracy of the p-value. If it is false, more resampling is needed to accurately estimate the p-value.
method.bin	a type of method to be used to compute the p-value.

Author(s)

Seunggeun Lee

References

Lee, S., Fuchsberger, C., Kim, S., Scott, L. (2015) An efficient resampling method for calibrating single and gene-based rare variant association analysis in case-control studies. *Biostatistics*, in press.

Examples

```
data(SKATBinary.example)
Z<-SKATBinary.example$Z


obj<-SKAT_Null_Model(y ~ x1 + x2, out_type="D", data=SKATBinary.example)
out = SKATBinary_Single(Z[,1], obj)

# p-value
out$p.value

# MAP
out$MAP

# method used to compute p-value (method.bin)
out$method.bin


#
# Use firth method to compute p-value
SKATBinary_Single(Z[,1], obj, method.bin="Firth")$p.value
```

SKAT_ChrX *SNP-set (Sequence) Kernel Association Test for X and Y chromosome variables*

Description

Test for association between a set of SNPS/genes in the X chromosome and continuous or dichotomous outcomes using the kernel machine.

Usage

```
SKAT_ChrX(Z, obj, is_X.inact =TRUE
, kernel = "linear.weighted", method="davies", weights.beta=c(1,25)
, weights = NULL, impute.method = "fixed", r.corr=0, is_check_genotype=TRUE
, is_dosage = FALSE, missing_cutoff=0.15, max_maf=1, estimate_MAF=1, SetID=NULL)

SKAT_ChrY(Z, obj, kernel = "linear.weighted", method="davies", weights.beta=c(1,25)
, weights = NULL, impute.method = "fixed", r.corr=0, is_check_genotype=TRUE
, is_dosage = FALSE, missing_cutoff=0.15, max_maf=1, estimate_MAF=1, SetID=NULL)
```

Arguments

Z	a numeric genotype matrix with each row as a different individual and each column as a separate gene/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.
obj	output object of the SKAT_Null_Model_ChrX function. For SKAT_ChrY, SKAT_Null_Model_ChrX function should be used with Model.Y=TRUE
is_X.inact	an indicator variable for the X-inactivation coding (default=TRUE). Male genotypes are coded as g=(0,2) when it is TRUE, and g=(0,1) when it is false.
kernel	a type of kernel (default= "linear.weighted").
method	a method to compute the p-value (default= "davies"). See SKAT page for details.
weights.beta	a numeric vector of parameters of beta weights. See SKAT page for details.
weights	a numeric vector of weights for the weighted kernels. See SKAT page for details.
impute.method	a method to impute missing genotypes (default= "fixed"). "fixed" imputes missing genotypes by assigning the mean genotype value (2p), and "bestguess" uses best guess genotype values.
r.corr	the ρ parameter for the compound symmetric kernel. See SKAT page for details.
is_check_genotype	a logical value indicating whether to check the validity of the genotype matrix Z (default= TRUE). See SKAT page for details.

`is_dosage` a logical value indicating whether the matrix Z is a dosage matrix. If it is TRUE, SKAT will ignore "is_check_genotype".

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

`max_maf` a cutoff of the maximum minor allele frequencies (MAF) (default=1, no cutoff). Any SNPs with MAF > cutoff will be excluded from the analysis.

`estimate_MAF` a numeric value indicating how to estimate MAFs for the weight calculation and the missing genotype imputation. See SKAT page for details.

`SetID` Internal use only.

Details

For details of parameters, please see SKAT page.

Value

`p.value` p-value of SKAT.

`p.value.resampling` p-values from resampled outcome. You can get it when you use obj from SKAT_Null_Model function with resampling. See the SKAT_Null_Model.

`p.value.noadj` p-value of SKAT without the small sample adjustment. It only appears when small sample adjustment is applied.

`p.value.noadj.resampling` p-values from resampled outcome without the small sample adjustment. It only appears when small sample adjustment is applied.

`pval.zero.msg` (only when p.value=0) text message that shows how small the p.value is. ex. "Pvalue < 1.000000e-60" when p.value is smaller than 10^{-60}

`Q` the test statistic of SKAT. It has NA when method="optimal.adj" or "optimal".

`param` estimated parameters of each method.

`param$Is_Converged` (only with method="davies") an indicator of the convergence. When 0 (not converged), "liu" method is used to compute p-value.

`param$n.marker` a number of SNPs in the genotype matrix

`param$n.marker.test` a number of SNPs used for the test. It can be different from param$n.marker when some markers are monomorphic or have higher missing rates than the missing_cutoff.

Author(s)

Clement Ma and Seunggeun Lee

Examples

```
data(SKAT.example.ChrX)
Z<-SKAT.example.ChrX$Z

############################################################
# Compute the P-value of SKAT

# binary trait
obj.x<-SKAT_Null_Model_ChrX(y ~ x1 +x2 + Gender,
  SexVar="Gender", out_type="D", data=SKAT.example.ChrX)

# SKAT
SKAT_ChrX(Z, obj.x, kernel = "linear.weighted")

# Burden
SKAT_ChrX(Z, obj.x, kernel = "linear.weighted", r.corr=1)

# SKAT-O
SKAT_ChrX(Z, obj.x, kernel = "linear.weighted", method="SKATO")

############################################################
# Fit the Y chromosome function
# In this example, since male has only one copy of X (and Y), we reuse X chromosome genotype matrix.

# binary trait
obj.x<-SKAT_Null_Model_ChrX(y ~ x1 +x2 + Gender,
  SexVar="Gender", out_type="D", Model.Y=TRUE, data=SKAT.example.ChrX)

SKAT_ChrY(Z, obj.x, kernel = "linear.weighted", method="SKATO")
```

SKAT_CommonRare *SKAT for the combined effect of common and rare variants*

Description

Sequence Kernel association test for the combined effect of common and rare variants.

Usage

```
SKAT_CommonRare(Z, obj, weights.beta.rare=c(1,25)
, weights.beta.common=c(0.5,0.5), weights=NULL
, method="C", r.corr.rare=0, r.corr.common=0, CommonRare_Cutoff=NULL
, test.type="Joint", is_dosage=FALSE, missing_cutoff=0.15
, estimate_MAF=1, SetID1=NULL)
```

```
SKAT_CommonRare.SSD.OneSet(SSD.INFO
, SetID, obj, ..., obj.SNPWeight=NULL)

SKAT_CommonRare.SSD.OneSet_SetIndex(SSD.INFO
, SetIndex, obj, ..., obj.SNPWeight=NULL )
```

Arguments

Z	a numeric genotype matrix with each row as a different individual and each column as a separate gene/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.
obj	an output object of the SKAT_Null_Model function.
weights.beta.rare	a numeric vector of parameters of beta weights for rare variants (default=c(1,25)).
weights.beta.common	a numeric vector of parameters of beta weights for common variants (default=c(0.5,0.5)).
weights	a numeric vector of weights for both common and rare variants. When it is NULL, the beta weight with the "weights.beta.rare" and "weights.beta.common" parameter are used. When method ="C", the coefficient to combine common and rare variants test statistics will be calculated after applying the weights to variants.
method	a method to combine common and rare variant effects (default="C"). "C" represents the combined sum test, and "A" represents the adaptive sum test. "AR" represents a different type of adaptive test in which common variants are projected over rare variants.
r.corr.rare	the ρ parameter for rare variants (default= 0). ρ =0 and 1 indicate SKAT and Burden test, respectively
r.corr.common	the ρ parameter for common variants (default= 0). ρ =0 and 1 indicate SKAT and Burden test, respectively
CommonRare_Cutoff	MAF cutoff for common vs rare variants (default=NULL). It should be a numeric value between 0 and 0.5, or NULL. When it is NULL, $1/\sqrt{2SampleSize}$ will be used.
test.type	a string to indicate test type (default="Joint"). "Joint" indicates the joint test of the combined effects of common and rare variants. "Rare.Only" and "Common.Only" will conduct test only with rare and common variants, respectively.
is_dosage	see SKAT
missing_cutoff	see SKAT

estimate_MAF	see SKAT
SetID1	internal use only
SSD.INFO	an SSD_INFO object returned from Open_SSD.
SetID	a character value of Set ID. You can find a set ID of each set from SetInfo object of SSD.INFO
SetIndex	a numeric value of Set index. You can find a set index of each set from SetInfo object of SSD.INFO
obj.SNPWeight	output object from Read_SNP_WeightFile (default=NULL). If NULL, the beta weight with the "weights.beta" parameter will be used.
...	furthuer arguments to be passed to "SKAT_CommonRare"

Details

The small sample adjustment for binary traits is not implemented for "A" and "AR".

Value

p.value	p-value.
p.value.resampling	p-values from resampled phenotypes. You can get it when you use obj from SKAT_Null_Model function with resampling. See the SKAT_Null_Model.
n.rare	the number of rare variants used for the test
n.common	the number of common variants used for the test
Cutoff	the MAF cut-off to divide common and rare variants
Q	the test statistic of SKAT. It has NA when method="A" or "AR".
param	estimated parameters of each method.
param$Is_Converged	an indicator of the convergence. 1 indicates the method is converged, and 0 indicates the method is not converged. When 0 (not converged), "liu.mod" method is used to compute p-value.
param$n.marker	a number of SNPs in the genotype matrix
param$n.marker.test	a number of SNPs used for the test. It can be different from param$n.marker when some markers are monomorphic or have higher missing rates than the missing_cutoff.
test.snp.mac	a vector of minor allele count (MAC) of the snps tested. The name is SNP-ID.

Author(s)

Seunggeun Lee

References

Ionita-Laza, I.*, Lee, S.*, Makarov, V., Buxbaum, J. Lin, X. (2013). Sequence kernel association tests for the combined effect of rare and common variants. *American Journal of Human Genetics*, 92, 841-853. * contributed equally.

Examples

```
data(SKAT.example)
Z<-SKAT.example$Z


# continuous trait
obj<-SKAT_Null_Model(y.c ~ X, out_type="C", data=SKAT.example)
SKAT_CommonRare(Z, obj)$p.value
SKAT_CommonRare(Z, obj, method="A")$p.value
SKAT_CommonRare(Z, obj, method="AR")$p.value


# dichotomous trait
obj<-SKAT_Null_Model(y.b ~ X, out_type="D", data=SKAT.example)

# Combined sum test in the manuscript (SKAT-C and Burden-C)
SKAT_CommonRare(Z, obj)$p.value
SKAT_CommonRare(Z, obj, r.corr.rare=1, r.corr.common=1 )$p.value


# Test only with common variant
SKAT_CommonRare(Z, obj, test.type="Common.Only")$p.value

# Test only with rare variant
SKAT_CommonRare(Z, obj, test.type="Rare.Only")$p.value


# Use CommonRare_Cutoff=0.01 instead of CommonRare_Cutoff = NULL
SKAT_CommonRare(Z, obj, CommonRare_Cutoff=0.01)$p.value

# Use custom weights; the first 10 variants have higher weights
weights<-rep(1,67); weights[1:10]<-2
SKAT_CommonRare(Z, obj, weights=weights)$p.value
```

SKAT_CommonRare_Robust

SNP set test (both common and rare variants) for binary traits with robust region-based methods

---

Description

This function computes p-values of robust burden test, SKAT, and SKAT-O for binary traits using SPA and ER.

Usage

```
SKAT_CommonRare_Robust(Z, obj, kernel = "linear.weighted"
```

```
, method="SKAT", r.corr=NULL,weights.beta.rare=c(1,25)
, weights.beta.common=c(0.5,0.5), weights = NULL
, CommonRare_Cutoff=NULL, impute.method = "bestguess"
, is_check_genotype=TRUE,is_dosage = FALSE, missing_cutoff=0.15
, max_maf=1, estimate_MAF=1)

SKAT_CommonRare_Robust.SSD.OneSet(SSD.INFO
, SetID, obj, ..., obj.SNPWeight=NULL)

SKAT_CommonRare_Robust.SSD.OneSet_SetIndex(SSD.INFO
, SetIndex, obj, ..., obj.SNPWeight=NULL )
```

Arguments

Z	a numeric genotype matrix with each row as a different individual and each column as a separate gene/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. Now we support both matrix and sparse matrix.
obj	output object from SKAT_Null_Model.
kernel	type of kernel (default= "linear.weighted"). The possible choices are "linear" and "linear.weighted".
method	type of gene based test (default= "SKAT"). The possible choices are "SKAT", "Burden" and "SKATO", which represents robust SKAT, Burden and SKAT-O tests, respectively.
r.corr	the ρ parameter for all variants. ρ =0 and 1 indicate SKAT and Burden test, respectively.
weights.beta.rare	a numeric vector of parameters of beta weights for rare variants. It is only used for weighted kernels. If you want to use your own weights, please specify the "weights" parameter.
weights.beta.common	a numeric vector of parameters of beta weights for common variants. It is only used for weighted kernels. If you want to use your own weights, please specify the "weights" parameter.
weights	a numeric vector of weights for the weighted kernels. See SKAT page for details.
CommonRare_Cutoff	MAF cutoff for common vs rare variants (default=NULL). It should be a numeric value between 0 and 0.5, or NULL. When it is NULL, $1/\sqrt{2SampleSize}$ will be used.
impute.method	a method to impute missing genotypes (default= "bestguess"). "bestguess" imputes missing genotypes as most likely values (0,1,2), "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).

`is_check_genotype`
: a logical value indicating whether to check the validity of the genotype matrix Z (default= TRUE). See SKAT page for details.

`is_dosage`
: a logical value indicating whether the matrix Z is a dosage matrix. If it is TRUE, SKAT will ignore "is_check_genotype".

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

`max_maf`
: a cutoff of the maximum minor allele frequencies (MAF) (default=1, no cutoff). Any SNPs with MAF > cutoff will be excluded from the analysis.

`estimate_MAF`
: a numeric value indicating how to estimate MAFs for the weight calculation and the missing genotype imputation. See SKAT page for details.

`SSD.INFO`
: an SSD_INFO object returned from Open_SSD.

`SetID`
: a character value of Set ID. You can find a set ID of each set from SetInfo object of SSD.INFO. In SKATBinary_Robust function, this parameter is for the internal use only.

`SetIndex`
: a numeric value of Set index. You can find a set index of each set from SetInfo object of SSD.INFO

`obj.SNPWeight`
: output object from Read_SNP_WeightFile (default=NULL). If NULL, the beta weight with the "weights.beta" parameter will be used.

`...`
: further arguments to be passed to "SKATBinary_Robust"

Value

`p.value`
: p-value. It will be the p-value based on robust methods.

`p.value_singlevariant`
: p-value for each single variant in this region-based test.

`param$n.marker`
: a number of SNPs in the genotype matrix.

`param$n.marker.test`
: a number of SNPs used for the test. It can be different from param$n.marker when some markers are monomorphic or have higher missing rates than the missing_cutoff.

`param$rho`
: the ρ parameter for all variants.

`n.common`
: A number of common markers used for the test.

`mac.common`
: The total minor allele count (MAC) of common markers used for the test.

`n.rare`
: A number of rare markers used for the test.

`mac.rare`
: The total minor allele count (MAC) of rare markers used for the test.

Author(s)

Zhangchen Zhao

References

Zhao, Z., Bi, W., Zhou, W., VandeHaar, P., Fritsche, L. G., & Lee, S. (2019). UK-Biobank Whole Exome Sequence Binary Phenome Analysis with Robust Region-based Rare Variant Test. *The American Journal of Human Genetics*, in press.

Examples

```
data(SKATBinary.example)
Z<-SKATBinary.example$Z

obj<-SKAT_Null_Model(y ~ x1 + x2, out_type="D", data=SKATBinary.example)

# run SKAT (default method) with Hybrid
out = SKAT_CommonRare_Robust(Z, obj)

# p-value
out$p.value

#
# Run burden and SKAT

SKAT_CommonRare_Robust(Z, obj, method="Burden")$p.value
SKAT_CommonRare_Robust(Z, obj, method="SKAT")$p.value
```

SKAT_NULL_emmaX *Get parameters and residuals from the null model with incorporating the kinship structure*

Description

Compute model parameters and residuals for SKAT with incorporating the kinship structure.

Usage

```
SKAT_NULL_emmaX (formula, data=NULL, K=NULL,
Kin.File=NULL, ngrids=100, llim=-10, ulim=10, esp=1e-10, Is.GetEigenResult=FALSE)
```

Arguments

formula	an object of class "formula": a symbolic description of the NULL model to be fitted.
data	an optional data frame containing the variables in the model (default=NULL). If it is NULL, the variables are taken from 'environment(formula)'
K	a kinship matrix. If K=NULL, the function reads Kin.File to get a kinship matrix.
Kin.File	an emmax-kin output file name. If K=NULL, the function reads this file.
ngrids	Number of grids to search for the optimal variance component

`llim`	Lower bound of log ratio of two variance components
`ulim`	Upper bound of log ratio of two variance components
`esp`	Tolerance of numerical precision error
`Is.GetEigenResult`	Return intermediate eigen-decomposition results

Details

The Emma package code was used to implement this function.

Resampling is not implemented.

Value

This function returns an object that has model parameters and residuals of the NULL model of no associations. After obtaining it, use SKAT function to carry out the association test.

Author(s)

Seunggeun Lee

Examples

```
data(SKAT.fam.example)

K = SKAT.fam.example$K
Z = SKAT.fam.example$Z
obj<-SKAT_NULL_emmaX(y ~ X, K=K, data=SKAT.fam.example)
SKAT(Z, obj)$p.value

# SKAT-O
SKAT(Z, obj, method="optimal.adj")$p.value
```

SKAT_Null_Model *Get parameters and residuals from the NULL model*

Description

Compute model parameters and residuals for SKAT. You also can obtain resampled residuals that can be used to compute resampling p-value or to control family-wise error rate.

Usage

```
SKAT_Null_Model(formula, data=NULL, out_type="C", n.Resampling=0
, type.Resampling="bootstrap", Adjustment=TRUE)

SKAT_Null_Model_ChrX(formula, SexVar, data=NULL, out_type="C", n.Resampling=0
, type.Resampling="bootstrap", Adjustment=TRUE, Model.Y=FALSE)
```

Arguments

formula	an object of class "formula": a symbolic description of the NULL model to be fitted.
data	an optional data frame containing the variables in the model (default=NULL). If it is NULL, the variables are taken from 'environment(formula)'
out_type	an indicator of the outcome type. "C" for the continuous outcome and "D" for the dichotomous outcome.
n.Resampling	a numeric value of the number of resampling (default=0). If you don't want resampling, please set n.Resampling=0.
type.Resampling	resampling methods (default="bootstrap"). see details.
Adjustment	If TRUE, a small sample adjustment will be applied when the sample size < 2000 and the trait is binary (default=TRUE). See details
SexVar	a sex variable name in "formula".
Model.Y	indicator variable whether the model will be also used for ChrY. It should be TRUE if you want to use SKAT_ChrY function

Details

There are 2 different methods to get resampled residuals. "bootstrap" conducts the parametric bootstrap to resample residuals under the NULL model with considering covariates. "bootstrap.fast" (only for binary traits) is a fast implementation of "bootstrap". If there is no covariate, "bootstrap" is equivalent to the permutation method.

When the trait is binary, the SKAT can produce conservative results when the sample size is small. To address this, we developed a small sample adjustment method, which adjusts asymptotic null distribution by estimating small sample moments. See also SKAT_Null_Model_MomentAdjust.

Since small sample adjustment uses random sampling to estimate the kurtosis of the test statistics, SKAT with the (kurtosis-based) small sample adjustment can yield slightly different p-values for each run. If you want to reproduce p-values, please set a seed number using set.seed function in R.

We recently developed more advanced methods to get p-values for binary traits, and the methods are implemented in SKATBinary. We recommend to use SKATBinary function instead of SKAT when your trait is binary.

Value

This function returns an object that has model parameters and residuals of the NULL model of no association between genetic variables and outcome phenotypes. After obtaining it, please use SKAT function to conduct the association test.

Author(s)

Seunggeun Lee

Examples

```
data(SKAT.example)
Z<-SKAT.example$Z
############################################################
# Compute the P-value of SKAT

# binary trait
obj<-SKAT_Null_Model(y.b ~ X, out_type="D", data=SKAT.example)
SKAT(Z, obj, kernel = "linear.weighted")$p.value


############################################################
#  When you have no covariate to adjust.

# binary trait
obj<-SKAT_Null_Model(y.b ~ 1, out_type="D", data=SKAT.example)
SKAT(Z, obj, kernel = "linear.weighted")$p.value



##########################################################
# Small sample adjustment
IDX<-c(1:100,1001:1100)

# With-adjustment
obj<-SKAT_Null_Model(y.b[IDX] ~ X[IDX,],out_type="D", data=SKAT.example)
SKAT(Z[IDX,], obj, kernel = "linear.weighted")$p.value

# Without-adjustment
obj<-SKAT_Null_Model(y.b[IDX] ~ X[IDX,],out_type="D", Adjustment=FALSE, data=SKAT.example)
SKAT(Z[IDX,], obj, kernel = "linear.weighted")$p.value

##########################################################
#  Use SKATBinary

SKATBinary(Z[IDX,], obj, kernel = "linear.weighted")$p.value
```

`SKAT_Null_Model_MomentAdjust` *Get parameters and residuals from the NULL model for small sample adjustment*

Description

Compute model parameters and residuals for SKAT with adjusting small sample moments when the trait is binary. You can also obtain resampled residuals that can be used to compute resampling p-value or to control family-wise error rate.

Usage

```
SKAT_Null_Model_MomentAdjust(formula, data=NULL, n.Resampling=0,
type.Resampling="bootstrap", is_kurtosis_adj=TRUE, n.Resampling.kurtosis=10000)
```

Arguments

`formula`	object of class "formula": a symbolic description of the NULL model to be fitted.
`data`	optional data frame containing the variables in the model (default=NULL). If it is NULL, the variables are taken from 'environment(formula)'
`n.Resampling`	a numeric value of the number of resampling (default=0). If you don't want resampling, please set n.Resampling=0.
`type.Resampling`	resampling methods (default="bootstrap"). see details.
`is_kurtosis_adj`	If TRUE, the kurtosis adjustment will be applied. The small sample kurtosis will be estimated using the resampled phenotypes.
`n.Resampling.kurtosis`	a numeric value of the number of resampling for kurtosis estimation (default=10000). If is_kurtosis_ad=FALSE, it will be ignored.

Details

When the trait is binary, the SKAT can produce conservative results when the sample size is small. To address this, we developed a small sample adjustment method, which adjust asymptotic null distribution by estimating small sample variance and kurtosis. The small smaple variance is estimated analytically, and the small sample kurtosis is estimated using the resampling approach.

There are 2 different methods to get resampled residuals. "bootstrap" conducts the parametric bootstrap to resample residuals under the NULL model with considering covariates. "bootstrap.fast" (only for binary traits) is a fast implementation of "bootstrap". If there is no covariate, "bootstrap" is equivalent to the permutation method.

Since the kurtosis is estimated using random samples, SKAT with the kurtosis-based small sample adjustment can yield slightly different p-values for each run. If you want to reproduce p-values, please set a seed number using set.seed function in R.

Value

This function returns an object that has model parameters and residuals of the NULL model of no association between genetic variants and outcome phenotypes. After obtaining it, please use SKAT function to conduct association tests.

Author(s)

Seunggeun Lee

Examples

```
data(SKAT.example)
Z<-SKAT.example$Z
############################################################
# Compute the P-value of SKAT

IDX<-c(1:100,1001:1100)

# binary trait
obj<-SKAT_Null_Model_MomentAdjust(y.b[IDX] ~ X[IDX,], data=SKAT.example)
SKAT(Z[IDX,], obj, kernel = "linear.weighted")$p.value
```

SSD_FILE_OPEN *Internal variables for SSD and functions for other packages*

Description

Internal variable for SSD and functions for other packages

Index