Package 'SKAT'

March 2, 2013

Type Package

Title SNP-set (Sequence) Kernel Association Test

Version 0.82

Date 2011-02-26

Author Seunggeun Lee, Larisa Miropolsky and Micheal Wu

Maintainer Seunggeun (Shawn) Lee <phila78@gmail.com>

Description Kernel based SNP set test

License GPL (>= 2)

Depends R (>= 2.13.0)

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-03-02 17:05:32

R topics documented:

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

Description

Close the 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 formated data files using user specified SNP sets. If you want to use plink formated data files, you must generate the SSD files first.

Usage

```
Generate_SSD_SetID(File.Bed, File.Bim, File.Fam, File.SetID,
 File.SSD, File.Info)
```

Arguments

`File.Bed`	the name of the binary ped file (BED).
`File.Bim`	the name of the binary map file (BIM).
`File.Fam`	the name of the FAM file (FAM).
`File.SetID`	the name of the Snp set ID file which defines SNP sets. The first column of the file must be Set ID, and the second column must be SNP ID. There should be no header!!
`File.SSD`	the name of the SSD file generated.
`File.Info`	the name of the SSD info file generated.

Details

The SetID file is 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 25 characters, otherwise, it will return error message.

The SSD file is a binary formated file with genotype information. The SSD info file is a text file. The first 6 rows have general information of data and SNP sets. The information of each set can be found from the 8th row.

Author(s)

Seunggeun Lee, Larisa Miropolsky

Get_Genotypes_SSD *Get Genotype data from SSD file*

Description

Read SSD file and return a genotype matrix.

Usage

```
Get_Genotypes_SSD(SSD_INFO, Set_Index)
```

Arguments

SSD_INFO	a SSD_INFO object returned from Open_SSD.
Set_Index	a numeric value of Set index. You can find a set index of each set from SetInfo object of SSD.INFO.

Value

The genotype matrix with n rows and m columns, where n is the number of samples, and m is the number of SNPs.

Author(s)

Seunggeun Lee, Larisa Miropolsky

`Get_Logistic_Weights` *Get the logistic weight*

Description

Get the logistic weights from either a genotype matrix (Z) or a vector of minor allele frequncies (MAF). You 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 of the logistic weight is

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

Value

It returns a vector of the logistic weight.

Author(s)

Seunggeun Lee

Examples

```
data(SKAT.example)
attach(SKAT.example)


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

# Use logistic weight
obj<-SKAT_Null_Model(y.c ~ X, out_type="C")
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

It computes required sample sizes by simple interpolation.

Value

A list object of the required sample sizes.

Author(s)

Seunggeun Lee

`Get_Resampling_Pvalue` *Compute the resampling p-value*

Description

To compute a resampling p-value using the resampled residuals. To use it, you need to obtain resampling residuals using SKAT_Null_Model, and then run SKAT.

Usage

```
Get_Resampling_Pvalue(obj)

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

Arguments

`obj`	a SKAT outcome object.
`p.value`	a numeric value of the SKAT p-value.
`p.value.resampling`	a vector of p-values of 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, and n1 is the number of resampled residual p-values smaller than the original sample p-value.
`is_smaller`	a logical value which indicates whether the resampling p-value should be smaller. If n1=0, then it has TRUE, otherwise it has FALSE.

Author(s)

Seunggeun Lee

Open_SSD *Open SNP set data file (SSD)*

Description

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

Usage

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

Arguments

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

Value

Open_SSD returns a list object of SSD.INFO which has set information.

Author(s)

Seunggeun Lee, Larisa Miropolsky

Power_Continuous *Power calculation, continuous traits*

Description

Compute an average power of SKAT 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 it has 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 mean 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, and thus there is 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 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.
BetaType	a function type of effect sizes (default= "Log"). "Log" indicates that effect size of each causal variant equals 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 of new class of kernels with compound symmetric correlation structure for genotype effects (default= 0). See details.

Details

By default it use 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 BetaType="Log", MaxBeta is the coeffecient value (β) of the causal SNP with 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 the 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 the 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 the 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 the proportion of causal variants, and p_2 is the 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 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 it has 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 mean 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, and thus there is no random selection of subregions.
`Prevalence`	a value of disease prevalence.
`Case.Prop`	a value of the proportion of case samples. 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 each causal variant equals 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 use 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 the odd ratio of the causal SNP with 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 the 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 the 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 the proportion of causal variants, and p_2 is the 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)
```

Read_Plink_FAM *Read a Plink FAM file*

Description

Read a Plink FAM file

Usage

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

Arguments

Filename	an 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. Please see the plink manual.

Value

A data frame of Family ID (FID), Individual ID (IID), Paternal ID (PID), Maternal ID(MID), Sex, and Phenotype.

Author(s)

Seunggeun Lee

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

Description

This function returns significant SNP sets after controlling for family wise error rate (FWER) using resampled residuals. To use it, you need to obtain resampling residuals using SKAT_Null_Model, and then conduct the SKAT repeatedly for all genes/SNP sets or use SKAT.SSD.All function.

Usage

```
Resampling_FWER(obj,FWER=0.05)

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

Arguments

`obj`	an object returned from SKAT.SSD.All function.
`P.value`	a vector of the SKAT p-value. If you test 100 genes, 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 you use the returned object from SKAT.SSD.all function, 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 has NULL value.
`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 outcomes using the kernel machine.

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 )

SKAT.SSD.OneSet(SSD.INFO, SetID, obj, ...)

SKAT.SSD.OneSet_SetIndex(SSD.INFO, SetIndex, obj, ... )
```

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, "optimal.adj" represents 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 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. 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"). "random" imputes missing genotypes by generating binomial(2,p) random variables (p is the MAF), and "fixed" imputes missing genotypes by assigning the mean genotype value (2p). If you use "random", you will have different p-values for different runs because imputed values are randomly assigned.

`r.corr` the ρ parameter of new class of kernels with compound symmetric correlation structure for genotype effects (default= 0). If you give a vector value, SKAT will conduct the optimal test. See details.

`is_check_genotype`
a logical value indicating whether to check the validity of the genotype matrix Z (default= TRUE). If you use non-SNP type data and want to run kernel machine test, please set it FALSE, otherwise you will get an error message. If you use SNP data or imputed data, please set it TRUE. If it is FALSE, and you use weighted kernels, the weights should be given through "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 cutoff will be excluded from the analysis.

`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

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

Details

There are pre-specified 6 types of 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. You can use one of them or your own kernel matrix as a parameter.

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

The kernel matrix of 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, open it first, 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 test statistic of SKAT, and Q_B is a test statistic of the weighted burden test. Thus, $\rho = 0$ 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, the optimal test will be conducted with adaptively seleting ρ from given r.corr. ρ should be a value between 0 and 1.

We slightly changed the implementation of SKAT-O to improve the tail probability. You can run it by using method="optimal.adj". It use a grid of eight points $\rho = (0, 0.1^2, 0.2^2, 0.3^2, 0.4^2, 0.5^2, 0.5, 1)$ to perform the search of the optimal ρ. If you want to use the original implementation of SKAT-), use method="optimal", which conducts SKAT-O with equal sized grid of 11 points (from 0 to 1). When the method is either "optimal.adj" or "optimal", the Q is NA.

If the true p.value is very small, you can have p.value=0 due to numerical reason. 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".

Value

`p.value` the p-value of SKAT.

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

`p.value.noadj.resampling`
: the p-value 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. 1 indicates the method is converged, and 0 indicates the method is not converged. 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)

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*, in press.

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(3), 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.

Madsen, B. E. and Browning S. R. (2009) A Groupwise Association Test for Rare Mutations Using a Weighted Sum Statistic. *PLOS Genetics*, 5: e1000384.

Examples

```
data(SKAT.example)
attach(SKAT.example)



############################################################
# Compute the P-value of SKAT with default Beta(1,25) Weights
# - without covariates

# continuous trait
obj<-SKAT_Null_Model(y.c ~ 1, out_type="C")
SKAT(Z, obj)$p.value

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


################################################
# Compute the P-value of SKAT with default Beta(1,25) Weights
```

```
# - with covariates

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

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

################################################
# Compute the P-value of SKAT with default Beta(1,25) Weights
# - Optimal Test

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


############################################################
# Compute the P-value of 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.haplotypes *Haplotype dataset for power calculation.*

Description

Haplotype dataset generated by the calibrated coalescent model 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 simulated using the calibration coalescent model with mimicking LD structure of European ancestry.

SNPInfo a dataframe object 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 conduct association tests with phenotypes and SNP sets in SSD file.

Usage

```
SKAT.SSD.All(SSD.INFO, obj, ...)
```

Arguments

SSD.INFO	an SSD_INFO object returned from Open_SSD.
obj	an output object of the SKAT_Null_Model function.
...	furthuer arguments to be passed to "SKAT".

Details

Please see SKAT for details.

Value

`results` the 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).

`P.value.Resampling` the matrix that contains p-values of resampled phenotypes.

Author(s)

Seunggeun Lee

SKAT_Null_Model *Get parameters and residuals from the H0 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)
```

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

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. If there is no covariate, "bootstrap" is equivalent to the permutation method. "perturbation" perturbs the residuals by multiplying mean zero and variance one normal random variable. The default method is "bootstrap".

We no longer provide "perturbation" method!

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

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)
attach(SKAT.example)

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

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

`SKAT_Null_Model_MomentAdjust` *Get parameters and residuals from the H0 model for small sample adjustment*

Description

Compute model parameters and residuals for SKAT with adjusting small sample moments when the trait is binary. 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_MomentAdjust(formula, data=NULL, n.Resampling=0,
type.Resampling="bootstrap", is_kurtosis_adj=TRUE, n.Resampling.kurtosis=10000)
```

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)'
`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 recently have 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. If there is no covariate, "bootstrap" is equivalent to the permutation method. "perturbation" perturbs the residuals by multiplying mean zero and variance one normal random variable. The default method is "bootstrap".

We no longer provide "perturbation" method!

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)
attach(SKAT.example)

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

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

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

SSD_FILE_OPEN *Interval variables for SSD files*

Description

Interval variables for SSD files

Index