Package 'famSKATRC'

October 13, 2022

Type Package

Title Family Sequence Kernel Association Test for Rare and Common Variants

Version 1.1.0

Author Khalid B. Kunji [aut, cre], Mohamad Saad [aut]

Maintainer Khalid B. Kunji <kkunji@hbku.edu.qa>

Repository CRAN

Description FamSKAT-RC is a family-based association kernel test for both rare and common variants. This test is general and several special cases are known as other methods: famSKAT, which only focuses on rare variants in family-based data, SKAT, which focuses on rare variants in population-based data (unrelated individuals), and SKAT-RC, which focuses on both rare and common variants in population-based data. When one applies famSKAT-RC and sets the value of phi to 1, famSKAT-RC becomes famSKAT. When one applies famSKAT-RC and set the value of phi to 1 and the kinship matrix to the identity matrix, famSKAT-RC becomes SKAT. When one applies famSKAT-RC and set the kinship matrix (fullkins) to the identity matrix (and phi is not equal to 1), famSKAT-RC becomes SKAT-RC. We also include a small sample synthetic pedigree to demonstrate the method with. For more details see Saad M and Wijsman EM (2014) <doi:10.1002/gepi.21844>.

License GPL (>= 3)

URL https://www.r-project.org, https://www.ncbi.nlm.nih.gov/pubmed/25132070

Imports CompQuadForm, kinship2, coxme, bdsmatrix

LinkingTo

Depends R (>= 3.4.1)

Encoding UTF-8

LazyData true

KeepSource TRUE

NeedsCompilation no

Date/Publication 2017-11-09 15:01:06 UTC

R topics documented:

`famSKATRC` *Family Sequence Kernel Association Test for Rare and Common Variants*

Description

FamSKAT-RC is a family-based association kernel test for both rare and common variants. This test is general and several special cases are known as other methods: famSKAT, which only focuses on rare variants in family-based data, SKAT, which focuses on rare variants in population-based data (unrelated individuals), and SKAT-RC, which focuses on both rare and common variants in population-based data. When one applies famSKAT-RC and sets the value of phi to 1, famSKAT-RC becomes famSKAT. When one applies famSKAT-RC and set the value of phi to 1 and the kinship matrix to the identity matrix, famSKAT-RC becomes SKAT. When one applies famSKAT-RC and set the kinship matrix (fullkins) to the identity matrix (and phi is not equal to 1), famSKAT-RC becomes SKAT-RC. We also include a small sample synthetic pedigree to demonstrate the method with. For more details see Saad M and Wijsman EM (2014) <doi:10.1002/gepi.21844>.

Author(s)

Khalid B. Kunji [aut, cre], Mohamad Saad [aut]

Maintainer: Khalid B. Kunji <kkunji@hbku.edu.qa>

References

Saad M and Wijsman EM (2014) Combining family- and population-based imputation data for association analysis of rare and common variants in large pedigrees. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4190076/

`famSKAT_RC` *famSKAT_RC*

Description

FamSKAT-RC is a family-based association kernel test for both rare and common variants.

Usage

```
famSKAT_RC (PHENO, genotypes, id, fullkins, covariates=NULL, sqrtweights_c,
            sqrtweights_r, binomialimpute=FALSE, acc=NULL, maf, phi)
```

Arguments

PHENO	The vector of the phenotype values. The missing values must be represented by NA.
genotypes	The genotype matrix. Its size should be N * P, where N is the number of individuals and P is the number of SNPs.
id	The vector of individual IDs to be included in the test. These IDs are present in the kinship matrix.
fullkins	The full kinship matrix that contains all individuals.
covariates	A matrix of possible covariates.
sqrtweights_c	The weight function to be assigned for common variants. An example is: function(maf) ifelse(maf>0, dbeta(maf,0.5,0.5), 0).
sqrtweights_r	The weight function to be assigned for rare variants. An example is: function(maf) ifelse(maf>0, dbeta(maf,1,25), 0).
binomialimpute	If TRUE, then impute missing genotypes using a binomial distribution (e.g. rbinom(10, 2, MAF), if 10 genotypes are missing).
acc	The accuracy used in the Davies approximation. For example, acc= 1e-06.
maf	A MAF threshold used to define rare and common variants (e.g. maf=0.01).
phi	This parameter indicates the contribution portion of rare variants. For example, a value of phi=0.5 means that the contribution of rare and common variants to the test is equaland a value of phi=1 means that only rare variants contribute to the test. You can vary the phi values as you wish and you can also provide a grid of different values (e.g. phi = c(0,0.2,0.5,0.9)). In this case, four p-values will be obtained, one for each value of phi.

Details

FamSKAT-RC is a family-based association kernel test for both rare and common variants. This test is general and several special cases are known as other methods: famSKAT, which only focuses on rare variants in family-based data, SKAT, which focuses on rare variants in population-based data (unrelated individuals), and SKAT-RC, which focuses on both rare and common variants in population-based data. When one applies famSKAT-RC and sets the value of phi to 1, famSKAT-RC becomes famSKAT. When one applies famSKAT-RC and set the value of phi to 1 and the kinship matrix to the identity matrix, famSKAT-RC becomes SKAT. When one applies famSKAT-RC and set the kinship matrix (fullkins) to the identity matrix (and phi is not equal to 1), famSKAT-RC becomes SKAT-RC.

Value

The test statistic p-value.

temp_pvalues	Some Description

Examples

```
library(kinship2)
sample.ped.geno <- process_data()
KIN = kinship(sample.ped.geno$IID, sample.ped.geno$FA, sample.ped.geno$MO)
IID = sample.ped.geno$IID
wuweights_r <- function(maf) ifelse(maf>0, dbeta(maf,1,25), 0)
wuweights_c <- function(maf) ifelse(maf>0, dbeta(maf,0.5,0.5), 0)
P_VALUES <- famSKAT_RC(PHENO=sample.ped.geno[,"Phenotype"],genotypes=as.matrix(
            sample.ped.geno[,7:ncol(sample.ped.geno)]), binomialimpute=TRUE,
            id=IID,fullkins=KIN,maf=0.05, sqrtweights_c=wuweights_c,
            sqrtweights_r=wuweights_r, phi = c(0,0.2,0.5,0.9))
print(P_VALUES)
```

`process_data` *Process Data*

Description

Processes the raw data included to produce data identical to the (also included) sample.ped.geno data set. This can be used for processing your own pedigrees with SNP data.

Usage

```
process_data(Data = read.table(system.file("extdata", "data",
                                           package = "famSKATRC"), header = TRUE))
```

Arguments

`Data`	A string, the path to the location of the data file you are processing, formatted as the included example, which can be loaded with: `read.table(system.file("extdata", "data", package = "famSKATRC"), header = TRUE)` and can be found in your filesystem at: system.file("extdata", "data", package = "famSKATRC")

Value

Returns the data frame with completed preprocessing changes for famSKATRC. Mainly reworking IDs so there are not duplicates.

See Also

famSKATRC

Examples

```
sample.ped.geno <- process_data()
## The function is currently defined as
function(Data = read.table(system.file("extdata", "data",
                                                 package = "famSKATRC"), header = TRUE))
{
```

```
  Data[ , "IID"] = paste(Data[ , "FID"]  , Data[ , "IID"]  ,sep=".")
  Data[Data[,"FA"]!=0 , "FA"] = paste(Data[Data[,"FA"]!=0 , "FID"], Data[Data[,"FA"]!=0,
                                                                         "FA"]  ,sep=".")
  Data[Data[,"FA"]!=0 , "MO"] = paste(Data[Data[,"FA"]!=0 , "FID"], Data[Data[,"FA"]!=0,
                                                                         "MO"]  ,sep=".")
  return(Data)
}
```

`sample.ped.geno` *Sample Pedigree Genotype Data*

Description

A sample pedigree file with SNP data, already processed by the process_data() function. The raw data is also included in the package.

Usage

```
data("sample.ped.geno")
```

Format

A data frame with 20 observations on the following 36 variables.

`FID` a numeric vector

`IID` a character vector

`FA` a character vector

`MO` a character vector

`SEX` a numeric vector

`Phenotype` a numeric vector

`rs1` a numeric vector

`rs2` a numeric vector

`rs3` a numeric vector

`rs4` a numeric vector

`rs5` a numeric vector

`rs6` a numeric vector

`rs7` a numeric vector

`rs8` a numeric vector

`rs9` a numeric vector

`rs10` a numeric vector

`rs11` a numeric vector

`rs12` a numeric vector

rs13 a numeric vector

rs14 a numeric vector

rs15 a numeric vector

rs16 a numeric vector

rs17 a numeric vector

rs18 a numeric vector

rs19 a numeric vector

rs20 a numeric vector

rs21 a numeric vector

rs22 a numeric vector

rs23 a numeric vector

rs24 a numeric vector

rs25 a numeric vector

rs26 a numeric vector

rs27 a numeric vector

rs28 a numeric vector

rs29 a numeric vector

rs30 a numeric vector

Details

A pedigree file with 20 individuals from two families. The first column is a family ID, the second an individual ID, the 3rd gives their father, the 4th their mother, and the 5th their sex. The 6th column gives their phenotype and columns 7 through 36 give their genotype, 30 SNP loci.

Source

This is synthetically generated data.

Examples

```
data(sample.ped.geno)
library(kinship2)
sample.ped.geno$FA[sample.ped.geno$FA == "0"] <- NA
sample.ped.geno$MO[sample.ped.geno$MO == "0"] <- NA
sample.ped.geno$Phenotype[sample.ped.geno$Phenotype >= 0] <- 1
sample.ped.geno$Phenotype[sample.ped.geno$Phenotype < 0] <- 0
ped.list <- pedigree(id = sample.ped.geno$IID, dadid = sample.ped.geno$FA,
                     momid = sample.ped.geno$MO, sex = sample.ped.geno$SEX,
                     famid = sample.ped.geno$FID,
                     affected = sample.ped.geno$Phenotype)
ped1 <- ped.list[1]
ped2 <- ped.list[2]
plot(ped1)
plot(ped2)
```

Index