Package 'ZIM4rv'

January 20, 2025

Type Package

Title Gene-based Association Tests of Zero-inflated Count Phenotype for Rare Variants

Version 0.1.1

Maintainer Xiaomin Liu <e0717571@u.nus.edu>

Description Gene-based association tests to model count data with excessive zeros and rare variants using zero-inflated Poisson/zero-inflated negative Binomial regression framework. This method was originally described by Fan, Sun, and Li in Genetic Epidemiology 46(1):73-86 <doi:10.1002/gepi.22438>.

License GPL-3

URL https://github.com/fanx0037/ZIM4rv

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends data.table, pscl, CompQuadForm, stats, SKAT, RNOmni, R (>= 3.5.0)

Suggests testthat (>= 3.0.0)

BugReports https://github.com/fanx0037/ZIM4rv/issues

NeedsCompilation no

Author Xiaomin Liu [aut, cre, cph]

Repository CRAN

Date/Publication 2024-12-19 16:10:07 UTC

Contents

---

cauchyp *cauchyp*

Description

Cauchy combination test (Cauchy-p)

This function combines p-values using Cauchy combination test for testing the joint genetic effect.

Usage

```
cauchyp(x)
```

Arguments

x	a numeric vector containing p-values

Value

a combined p-value indicating the joint effect

Ex1_genedata *An example dataset of genedata*

Description

Small, artificially generated toy data set that provides artificial information of genotypes for 200 individuals on 3 rs locations to illustrate the analysis with the use of the package.

Usage

```
data(Ex1_genedata)
```

Format

An object of class `"data.frame"`

FID Family IDs

IID Individual IDs

rs1 Genotype code for rs1

rs2 Genotype code for rs2

rs3 Genotype code for rs3

References

This data set was artificially created and modified for the ZIM4rv package.

Examples

```
data(Ex1_genedata)
head(Ex1_genedata)
```

Ex1_phenodata *An example dataset of phenodata*

Description

Small, artificially generated toy data set that provides artificial information of count phenotypes and covariates for 200 individuals to illustrate the analysis with the use of the package.

Usage

```
data(Ex1_phenodata)
```

Format

An object of class "data.frame"

FID Family IDs

IID Individual IDs

count Zero-inflated count phenotypes

educ Covariate education years

sex Covariate sex

PC1 The first principal component

PC2 The second principal component

PC3 The third principal component

References

This data set was artificially created and modified for the ZIM4rv package.

Examples

```
data(Ex1_phenodata)
head(Ex1_phenodata)
```

Ex2_covar *An example dataset of covariate file*

Description

Small, artificially generated toy data set that provides artificial information of covariates for 15 individuals to illustrate the pre-processing with the use of the package.

Usage

```
data(Ex2_covar)
```

Format

An object of class "data.frame" listing IDs and covariates separately

References

This data set was artificially created and modified for the ZIM4rv package.

Examples

```
data(Ex2_covar)
head(Ex2_covar)
```

Ex2_dosage *An example dataset of dosage file*

Description

Small, artificially generated toy data set that provides artificial information of dosage for 15 individuals to illustrate the pre-processing with the use of the package.

Usage

```
data(Ex2_dosage)
```

Format

An object of .dosage file

References

This data set was artificially created and modified for the ZIM4rv package.

Examples

```
data(Ex2_dosage)
head(Ex2_dosage)
```

Ex2_fam *An example dataset of .fam file*

Description

Small, artificially generated toy data set that provides artificial information of .fam file for 15 individuals to illustrate the pre-processing with the use of the package.

Usage

```
data(Ex2_fam)
```

Format

An object of standard .fam file

References

This data set was artificially created and modified for the ZIM4rv package.

Examples

```
data(Ex2_fam)
head(Ex2_fam)
```

Ex2_pheno *An example dataset of pheno file*

Description

Small, artificially generated toy data set that provides artificial information of phenotypes for 15 individuals to illustrate the pre-processing with the use of the package.

Usage

```
data(Ex2_pheno)
```

Format

An object of class "data.frame" listing IDs and phenotypes separately

References

This data set was artificially created and modified for the ZIM4rv package.

Examples

```
data(Ex2_pheno)
head(Ex2_pheno)
```

Ex2_region *An example dataset of genetic region file*

Description

Small, artificially generated toy data set that provides artificial information of 3 genetic regions to illustrate the pre-processing with the use of the package.

Usage

```
data(Ex2_region)
```

Format

An object of class "data.frame" listing genetic regions where each row contains chromosome, basepairs and the name of genetic region respectively

References

This data set was artificially created and modified for the ZIM4rv package.

Examples

```
data(Ex2_region)
head(Ex2_region)
```

`phi_lambda_hat` *phi_lambda_hat*

Description

Estimation of phi_hat and lambda_hat for ZIP model

This function gives the estimation of 2 parameters phi and lambda for each subject under the null hypothesis.

Usage

```
phi_lambda_hat(simud)
```

Arguments

`simud`	a data frame containing a phenotype named y and covariates

Details

This function first fits zero-inflated Poisson regression of phenotype y on the covariates only to obtain the estimates of regression coefficients and then compute the estimations of phi and lambda.

Value

a list of 2 estimations of parameters for each subject

See Also

`zeroinfl`

phi_mu_hat4zinb *phi_mu_hat4zinb*

Description

Estimation of phi_hat, mu_hat and alpha_hat for ZINB model

This function gives the estimation of three parameters phi, mu and alpha in ZINB model for each subject under the null hypothesis.

Usage

```
phi_mu_hat4zinb(simud)
```

Arguments

simud	a data frame containing a phenotype named y and covariates

Details

This function first fits zero-inflated negative binomial regression of phenotype y on the covariates only to obtain the estimates of regression coefficients and inverse dispersion and then compute the estimations of phi, mu and alpha.

Value

a list of 3 estimations of parameters for each subject

See Also

zeroinfl

preprocess_genedata *preprocess_genedata*

Description

Preprocess genotype files in PLINK format

This function converts PLINK format files into data frames containing genotypes information in proper format for the model fitting and testing.

Usage

```
preprocess_genedata(fam_file, dosage_file, region_file, gene_name)
```

Arguments

fam_file	.fam file in PLINK format
dosage_file	a dosage file includes dosage information of each variant for all individuals
region_file	a file listing genetic regions where each row contains chromosome, basepairs and the name of genetic region respectively
gene_name	a character string of the name of a gene, e.g."CEPT". The name is case-sensitive.

Value

a data frame containing genotypes for all individuals in the required format for model fitting and testing

Examples

```
data(Ex2_fam)
data(Ex2_dosage)
data(Ex2_region)
preprocess_genedata(Ex2_fam,Ex2_dosage,Ex2_region,"r2")
```

---

preprocess_phenodata *preprocess_phenodata*

Description

Preprocess phenotype files in PLINK format

This function converts PLINK format files into data frames containing phenotypes and covariates information in proper format for the model fitting and testing.

Usage

```
preprocess_phenodata(pheno_file, cov_file)
```

Arguments

pheno_file	phenotype file in PLINK format
cov_file	covariate file in PLINK format

Value

a data frame containing phenotypes and covariates respectively for all individuals in the required format for model fitting and testing

Examples

```
data(Ex2_pheno)
data(Ex2_covar)
preprocess_phenodata(Ex2_pheno,Ex2_covar)
```

p_burden_single *p_burden_single*

Description

Compute the p-value for the burden test

This function takes a vector of weights, a data frame of rare variants and a matrix of Score statistics produced by `U_fi_lmd` for ZIP model or `U_phi_mu4zinb` for ZINB model to compute the p-value for the burden test.

Usage

```
p_burden_single(wt, G_rare, s)
```

Arguments

`wt`	a numeric vector containing weights for all variants
`G_rare`	a data frame containing data of rare variants
`s`	a matrix of the score statistics for each variant from each subject

Value

the p-value for the burden test

---

p_kernel_single *p_kernel_single*

Description

Compute the p-value for the kernel test

This function takes a diagonal matrix of weights, a data frame of rare variants and a matrix of Score statistics produced by `U_fi_lmd` for ZIP model or `U_phi_mu4zinb` for ZINB model to compute the p-value for the kernel test.

Usage

```
p_kernel_single(wt_matrix2, G_rare, s)
```

Arguments

`wt_matrix2`	a diagonal matrix containing the squared weights for all variants
`G_rare`	a data frame containing data of rare variants
`s`	a matrix of the score statistics for each variant from each subject

Value

the p-value for the kernel test (ZIP-k)

`U_fi_lmd` *U_fi_lmd*

Description

Compute Score statistics for ZIP model

This function takes the estimations of phi and lambda produced by the `phi_lambda_hat` and computes the score statistics under the null hypothesis.

Usage

```
U_fi_lmd(simudata, G_rare)
```

Arguments

`simudata`	a data frame containing a phenotype named y and covariates
`G_rare`	a data frame containing data of rare variants with the same subject order as in simudata

Value

a list of 2 matrice of the score statistics for each variant from each subject

---

`U_phi_mu4zinb` *U_phi_mu4zinb*

Description

Compute score statistics for ZINB model

This function takes the estimations of phi and lambda produced by the `phi_lambda_hat4negbin` and computes the score statistics for ZINB model under the null hypothesis.

Usage

```
U_phi_mu4zinb(simudata, G_rare)
```

Arguments

`simudata`	a data frame containing a phenotype named y and covariates
`G_rare`	a data frame containing data of rare variants with the same subject order as in simudata

Value

a list of 2 matrice of the score statistics for each variant from each subject

---

`vuong_test` *vuong_test*

Description

Vuong's test

This function performs Vuong's test, a likelihood ratio test for model selection and non-nested hypotheses. This function is for model selection between zero-inflated Poisson model and zero-inflated negative binomial model.

Usage

```
vuong_test(phenodata)
```

Arguments

`phenodata`	a data frame containing family and individual IDs for all objects as well as zero-inflated counts as a phenotype and a set of covariates. Each row represents a different individual. The first two columns are Family ID (FID) and Individual ID (IID) respectively. There must be one and only one phenotype in the third column and the phenotype have to be zero-inflated count data which should be non-negative integers, e.g. neuritic plaque counts. Each of the rest of columns represents a different covariate, e.g. age, sex, etc.

Value

nothing returned, prints a table of 3 test statistics and p values, and exits silently.

`zimfrv` *zimfrv*

Description

Gene-based association tests to model zero-inflated count data

This function performs gene-based association tests between a set of SNPs/genes and zero-inflated count data using ZIP regression or ZINB regression or two-stage SKAT model framework.

Usage

```
zimfrv(
  phenodata,
  genedata,
  genename = "NA",
  weights = "Equal",
  missing_cutoff = 0.15,
  max_maf = 1,
  model = "zip"
)
```

Arguments

phenodata	a data frame containing family and individual IDs for all objects as well as zero-inflated counts as a phenotype and a set of covariates. Each row represents a different individual. The first two columns are Family ID (FID) and Individual ID (IID) respectively. There must be one and only one phenotype in the third column and the phenotype have to be zero-inflated count data which should be non-negative integers, e.g. neuritic plaque counts. Each of the rest of columns represents a different covariate, e.g. age, sex, etc.
genedata	a data frame containing family and individual IDs for all objects as well as numeric genotype data. Each row represents a different individual. The first two columns are Family ID (FID) and Individual ID (IID) respectively. Each of the rest columns represents a seperate gene/SNP marker. The genotype should be coded as 0, 1, 2 and NA for AA, Aa, aa and missing. Both of Family ID (FID) and Individual ID (IID) for each row in the 'genedata' derived from the PLINK formatted files should be in the same order as in the 'phenodata'. The number of rows in 'genedata' should be equal to the number of rows in 'phenodata'.
genename	a character string of the name of a gene, e.g. "CETP". The name is case-sensitive.
weights	a character string of pre-specified variant weighting schemes (default="Equal"). "Equal" represents no weight, "MadsenBrowning" represents the Madsen and Browning (2009) weight, "Beta" represents the Beta weight.
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.
model	character specification of zero-inflated count model family (default="zip"). "zip" represents Zero-Inflated Poisson model, "zinb" represents Zero-Inflated Negative Binomial model, "skat" represents the two-stage Sequence Kernel Association Test method.

Value

a list of 10 items including the name of gene, the number of rare variants in the genetic region, the kind of method used for modeling, and individual p-values of gene-based association tests (burden test and kernel test for both parameters) and combined p-values using Cauchy combination test.

`GeneName` the name of gene.

`No.Var` the number of rare variants in the gene.

`Method` the method used to compute the p-values.

`p.value_pi_burden`
single p-value for parameter π using burden test.

`p.value_lambda_burden` / `p.value_mu_burden`
single p-value for parameter λ or μ using burden test.

`p.value_pi_kernel`
single p-value for parameter π using kernel test.

`p.value_lambda_kernel` / `p.value_mu_kernel`
single p-value for parameter λ or μ using kernel test.

`p.value_pi_combined`
Combined p-value of testing parameter π from both burden and kernel test using Cauchy combination test.

`p.value_lambda_combined` / `p.value_mu_combined`
Combined p-value of testing parameter λ or μ from both burden and kernel test using Cauchy combination test.

`p.value_overall`
Combined p-value of testing the overall association using Cauchy combination test.

References

Fan, Q., Sun, S., & Li, Y.-J. (2021). Precisely modeling zero-inflated count phenotype for rare variants. Genetic Epidemiology, 1–14.

Examples

```
data(Ex1_phenodata)
data(Ex1_genedata)
zimfrv(Ex1_phenodata,Ex1_genedata,weights = "Beta",max_maf = 0.02,model="zinb")
```

Index