

# Package ‘FSTpackage’

May 7, 2026

**Type** Package

**Title** Unified Sequence-Based Association Tests Allowing for Multiple Functional Annotation Scores

**Version** 0.1

**Date** 2016-12-14

**Author** Zihuai He

**Maintainer** Zihuai He <zihuai@umich.edu>

**Description** Functions for sequencing studies allowing for multiple functional annotation scores. Score type tests and an efficient perturbation method are used for individual gene/large gene-set/genome wide analysis. Only summary statistics are needed.

**License** GPL-3

**Depends** R (>= 2.10), CompQuadForm, SKAT, Matrix, MASS, mvtnorm

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-06-27 16:00:43 UTC

## Contents

FST.example . . . . .	2
FST.GeneSet.test . . . . .	2
FST.prelim . . . . .	3
FST.SummaryStat.test . . . . .	4
FST.test . . . . .	5

<b>Index</b>	<b>8</b>
--------------	----------

---

FST.example	<i>Data example for FSTest (tests for genetic association allowing for multiple functional annotation scores)</i>
-------------	-------------------------------------------------------------------------------------------------------------------

---

### Description

The dataset contains outcome variable Y, covariate X, genotype data G, functional scores Z and gene-set ID for each variable GeneSetID.

### Usage

```
data(FST.example)
```

---

FST.GeneSet.test	<i>Test the association between an quantitative/dichotomous outcome variable and a large gene-set by a score type test allowing for multiple functional annotation scores.</i>
------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

### Description

Once the preliminary work is done using "FST.prelim()", this function tests a specific gene.

### Usage

```
FST.GeneSet.test(result.prelim,G,Z,GeneSetID,Gsub.id=NULL,weights=NULL,
B=5000,impute.method='fixed')
```

### Arguments

result.prelim	The output of function "FST.prelim()"
G	Genetic variants in the target gene, an n*p matrix where n is the subject ID and p is the total number of genetic variables. Note that the number of rows in G should be same as the number of subjects. ***The column name should be the variable name, in order to be matched with the GeneSetID.
Z	Functional annotation scores, an p*q matrix where p is the total number of genetic variables and q is the number of functional annotation scores. Note that the first column in Z should be all 1 if the users want the original weights of SKAT/burden test to be included.
GeneSetID	A p*2 matrix indicating the genes in which the variables are located, where the first column is the genes' name and the second column is the variables' name.
Gsub.id	The subject id corresponding to the genotype matrix, an n dimensional vector. This is in order to match the phenotype and genotype matrix. The default is NULL, where the order is assumed to be matched with Y, X and time.

weights	A numeric vector of weights for genetic variants (The length should be same as the number of genetic variants in the set). These weights are usually based on minor allele frequencies. The default is NULL, where the beta(1,25) weights are applied.
B	Number of Bootstrap replicates. The default is 5000.
impute.method	Choose the imputation method when there is missing genotype. Can be "random", "fixed" or "bestguess". Given the estimated allele frequency, "random" simulates the genotype from binomial distribution; "fixed" uses the genotype expectation; "Best guess" uses the genotype with highest probability.

### Value

n.marker	number of heterozygous SNPs in the SNP set.
p.value	P-value of the set based generalized score type test.

### Examples

```
## FST.prelim does the preliminary data management.
# Input: Y, X (covariates)
## FST.test tests a region.
# Input: G (genetic variants), Z (functional annotation scores) and result of FST.prelim

library(FSTpackage)

# Load data example
# Y: outcomes, n by 1 matrix where n is the total number of observations
# X: covariates, n by d matrix
# G: genotype matrix, n by p matrix where n is the total number of subjects
# Z: functional annotation matrix, p by q matrix

data(FST.example)
Y<-FST.example$Y;X<-FST.example$X;G<-FST.example$G;Z<-FST.example$Z;GeneSetID<-FST.example$GeneSetID

# Preliminary data management
result.prelim<-FST.prelim(Y,X=X,out_type='D')

# test with 5000 bootstrap replicates
result<-FST.GeneSet.test(result.prelim,G,Z,GeneSetID,B=5000)
```

---

FST.prelim

*The preliminary data management for FST (functional score tests)*


---

### Description

Before testing a specific gene using a score type test, this function does the preliminary data management, such as fitting the model under the null hypothesis.

**Usage**

```
FST.prelim(Y, X=NULL, id=NULL, out_type="C")
```

**Arguments**

Y	The outcome variable, an n*1 matrix where n is the total number of observations
X	An n*d covariates matrix where d is the total number of covariates.
id	The subject id. This is used to match the genotype matrix. The default is NULL, where the a matched phenotype and genotype matrix is assumed.
out_type	Type of outcome variable. Can be either "C" for continuous or "D" for dichotomous. The default is "C".

**Value**

It returns a list used for function FST.test().

**Examples**

```
library(FSTpackage)

# Load data example
# Y: outcomes, n by 1 matrix where n is the total number of observations
# X: covariates, n by d matrix
# G: genotype matrix, n by p matrix where n is the total number of subjects
# Z: functional annotation matrix, p by q matrix

data(FST.example)
Y<-FST.example$Y;X<-FST.example$X;G<-FST.example$G

# Preliminary data management
result.prelim<-FST.prelim(Y,X=X)
```

---

FST.SummaryStat.test *Using summary statistics to test the association between an quantitative/dichotomous outcome variable and a gene by a score type test allowing for multiple functional annotation scores.*

---

**Description**

This function tests a specific gene using summary statistics (score vector and its covariance matrix)

**Usage**

```
FST.SummaryStat.test(score,Sigma,Z,weights,B=5000)
```

**Arguments**

score	The score vector of length p, where p is the total number of genetic variables.
Sigma	The p*p covariance matrix of the score vector
Z	Functional annotation scores, an p*q matrix where p is the total number of genetic variables and q is the number of functional annotation scores. Note that the first column in Z should be all 1 if the users want the original weights of SKAT/burden test to be included.
weights	A numeric vector of weights for genetic variants (The length should be same as the number of genetic variants in the set.). These weights are usually based on minor allele frequencies.
B	Number of Bootstrap replicates. The default is 5000.

**Value**

p.value	P-value of the set based generalized score type test.
---------	-------------------------------------------------------

**Examples**

```
## FST.SummaryStat.test tests a region.
# Input: score (a score vector), Sigma (the covariance matrix of the score vector)

library(FSTpackage)

data(FST.example)
score<-FST.example$score;Sigma<-FST.example$Sigma;Z<-FST.example$Z;weights<-FST.example$weights

# test with 5000 bootstrap replicates
result<-FST.SummaryStat.test(score,Sigma,Z,weights,B=5000)
```

---

FST.test	<i>Test the association between an quantitative/dichotomous outcome variable and a gene by a score type test allowing for multiple functional annotation scores.</i>
----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

**Description**

Once the preliminary work is done using "FST.prelim()", this function tests a specific gene.

**Usage**

```
FST.test(result.prelim,G,Z,Gsub.id=NULL,weights=NULL,B=5000,impute.method='fixed')
```

**Arguments**

<code>result.prelim</code>	The output of function "FST.prelim()"
<code>G</code>	Genetic variants in the target gene, an $n \times p$ matrix where $n$ is the subject ID and $p$ is the total number of genetic variables. Note that the number of rows in $G$ should be same as the number of subjects.
<code>Z</code>	Functional annotation scores, an $p \times q$ matrix where $p$ is the total number of genetic variables and $q$ is the number of functional annotation scores. Note that the first column in $Z$ should be all 1 if the users want the original weights of SKAT/burden test to be included.
<code>Gsub.id</code>	The subject id corresponding to the genotype matrix, an $n$ dimensional vector. This is in order to match the phenotype and genotype matrix. The default is NULL, where the order is assumed to be matched with $Y$ , $X$ and time.
<code>weights</code>	A numeric vector of weights for genetic variants (The length should be same as the number of genetic variants in the set.). These weights are usually based on minor allele frequencies. The default is NULL, where the beta(1,25) weights are applied.
<code>B</code>	Number of Bootstrap replicates. The default is 5000.
<code>impute.method</code>	Choose the imputation method when there is missing genotype. Can be "random", "fixed" or "bestguess". Given the estimated allele frequency, "random" simulates the genotype from binomial distribution; "fixed" uses the genotype expectation; "Best guess" uses the genotype with highest probability.

**Value**

<code>n.marker</code>	number of heterozygous SNPs in the SNP set.
<code>p.value</code>	P-value of the set based generalized score type test.

**Examples**

```
## FST.prelim does the preliminary data management.
# Input: Y, X (covariates)
## FST.test tests a region.
# Input: G (genetic variants), Z (functional annotation scores) and result of FST.prelim

library(FSTpackage)

# Load data example
# Y: outcomes, n by 1 matrix where n is the total number of observations
# X: covariates, n by d matrix
# G: genotype matrix, n by p matrix where n is the total number of subjects
# Z: functional annotation matrix, p by q matrix

data(FST.example)
Y<-FST.example$Y;X<-FST.example$X;G<-FST.example$G;Z<-FST.example$Z

# Preliminary data management
result.prelim<-FST.prelim(Y,X=X,out_type='D')
```

*FST.test*

7

```
# test with 5000 bootstrap replicates  
result<-FST.test(result.prelim,G,Z,B=5000)
```

# Index

- \* **datasets**

- FST.example, 2

- \* **preliminary work**

- FST.prelim, 3

- \* **test**

- FST.GeneSet.test, 2

- FST.SummaryStat.test, 4

- FST.test, 5

FST.example, 2

FST.GeneSet.test, 2

FST.prelim, 3

FST.SummaryStat.test, 4

FST.test, 5