

# Package ‘FILEST’

June 10, 2018

**Type** Package

**Title** Fine-Level Structure Simulator

**Version** 1.0.3

**Description** A population genetic simulator, which is able to generate synthetic datasets for single-nucleotide polymorphisms (SNP) for multiple populations. The genetic distances among populations can be set according to the Fixation Index (Fst) as explained in Balding and Nichols (1995) <doi:10.1007/BF01441146>. This tool is able to simulate outlying individuals and missing SNPs can be specified. For Genome-wide association study (GWAS), disease status can be set in desired level according risk ratio.

**Depends** R (>= 3.2.4.0)

**License** GPL-2 | GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Imports** KRIS (>= 1.1.1),rARPACK,grDevices,stats,utils

**Suggests** testthat

**BugReports** <https://gitlab.com/kris.ccp/filest/issues>

**URL** <https://gitlab.com/kris.ccp/filest>

**NeedsCompilation** no

**Author** Kridsakorn Chaichoompu [aut, cre],  
Kristel Van Steen [aut],  
Fentaw Abegaz [aut]

**Maintainer** Kridsakorn Chaichoompu <kridsakorn@biostatgen.org>

**Repository** CRAN

**Date/Publication** 2018-06-10 16:12:58 UTC

## R topics documented:

cbind_bigmatrix . . . . .	2
demo.filest . . . . .	3
filest . . . . .	3
rbind_bigmatrix . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

cbind_bigmatrix	<i>Combine two matrices by column for big data, internally used for parallelization</i>
-----------------	---

---

### Description

Combine two matrices by column for big data, internally used for parallelization

### Usage

```
cbind_bigmatrix(a, b)
```

### Arguments

a	The first matrix
b	The second matrix

### Value

The combined matrix by column

### See Also

[rbind\\_bigmatrix](#)

### Examples

```
X <- matrix(c(1,2,0,1,2,2,1,2,0,0,1,2,1,2,2,2),ncol=4)
Y <- matrix(c(2,1,1,0,1,0,0,1,1,2,2,0,0,1,1,0),ncol=4)
Z <- cbind_bigmatrix(X,Y)
print(Z)
```

---

`demo.filest`*Demonstration the filest function*

---

**Description**

This function generates the setting file and demonstrate how to use `filest`.

**Usage**

```
demo.filest()
```

**Value**

The output directory

**Examples**

```
#To run this function, simply call demo.filest()

demo.filest()
```

---

`filest`*Simulate data for multiple populations*

---

**Description**

The output files are saved to the specified directory according to `out`.

**Usage**

```
filest(setting, out, thread = 1)
```

**Arguments**

<code>setting</code>	An absolute path to a setting file
<code>out</code>	An absolute path for output files
<code>thread</code>	A number to specify a maximum thread to be run in parallel

## Details

This function takes the specific input file containing the settings for simulations. It allows multiple settings for several simulation within one file. The simulation-setting file must be a text file. The line started with "-" indicates the parameters for simulation, and the line started with "#" are comments. Empty lines are allowed in the setting file. The parameters in the setting file are listed below:

- --setting A name of setting
- --population A list that indicates the numbers of population separated by comma
- --fst A list that indicates the Fst values separated by comma. Each Fst value represents a genetic distance of that particular population and the first population. The Fst values for the first population and the second population should be the same values, otherwise they will be summed up and divided by two.
- --case A list that indicates the ratio values of cases separated by comma
- --outlier A list that indicates the logical values (0/1) whether that population are outliers, separated by comma
- --marker A number of SNPs
- --replicate A number of replicates
- --riskratio A number of replicates
- --no.case.snp A number of case SNPs
- --pc A logical value (TRUE/FALSE) whether PCs will be calculated.
- --fulloutput A logical value (TRUE/FALSE) whether all information will be exported.

## Value

NULL if done successfully. NA if output directory can't be created.

## Examples

```
#Check and run the demo from demo.filest()

demo.filest()

#Here is the code for demo.filest()
txt <- "--setting=example1\n"
txt <- paste0(txt, "--population=500,500\n")
txt <- paste0(txt, "--fst=0.005,0.005\n")
txt <- paste0(txt, "--case=0,0\n")
txt <- paste0(txt, "--outlier=0,0\n")
txt <- paste0(txt, "--marker=3000\n")
txt <- paste0(txt, "--replicate=1\n")
txt <- paste0(txt, "--riskratio=1\n")
txt <- paste0(txt, "--no.case.snp=0\n")
txt <- paste0(txt, "--pc=TRUE\n")
txt <- paste0(txt, "--missing=0\n")
txt <- paste0(txt, "--fulloutput=TRUE\n")
```

```
outdir <- file.path(tempdir())

settingfile <- file.path(outdir, "example1.txt")
fo <- file(settingfile,"w")
for (i in txt){ write(i,fo)}
close(fo)

filest(setting = settingfile, out = outdir, thread = 1)
```

---

rbind_bigmatrix	<i>Combine two matrices by row for big data, internally used for parallelization</i>
-----------------	--

---

### Description

Combine two matrices by row for big data, internally used for parallelization

### Usage

```
rbind_bigmatrix(a, b)
```

### Arguments

a	The first matrix
b	The second matrix

### Value

The combined matrix by row

### See Also

[cbind\\_bigmatrix](#)

### Examples

```
X <- matrix(c(1,2,0,1,2,2,1,2,0,0,1,2,1,2,2,2),ncol=4)
Y <- matrix(c(2,1,1,0,1,0,0,1,1,2,2,0,0,1,1,0),ncol=4)
Z <- rbind_bigmatrix(X,Y)
print(Z)
```

# Index

`cbind_bigmatrix`, 2, 5

`demo.filest`, 3

`filest`, 3, 3

`rbind_bigmatrix`, 2, 5